unclassified
SECURITY CLASSIFICATION OF THIS PAGE

ADA208838

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Interface Foundation of North America, Inc. / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: ONR
6c. ADDRESS (City, State, and ZIP Code): P.O. Box 7460, Fairfax Station, Virginia 22039-7460	7b. ADDRESS (City, State, and ZIP Code): ONR Code 1111, 800 North Quincy Street, Arlington, VA 22217-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: ONR / 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-88-J-1049
8c. ADDRESS (City, State, and ZIP Code): ONR Code 1111, 800 North Quincy Street, Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
Computing Science and Statistics: Proceedings of the 20th Symposium on the Interface (U)

12. PERSONAL AUTHOR(S)
Edward J. Wegman, Donald T. Gantz, John J. Miller

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Proceedings	FROM 4/1/88 TO 8/20/88	1989 March 15	860

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computing Science, Statistics, Computational Statistics Computationally Intensive, Bootstrapping, Parallel Computing, Supercomputing, Neural Networks

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The 20th Symposium on the Interface: Computing Science and Statistics was held 20-23 April, 1988 in Reston, VA. The theme was computationally intensive methods in statistics. Some 60 invited papers, 128 contributed papers were presented to 425 attendees. There was a special focus on young investigators. Sessions were organized into some 49 technical sessions. This report presents papers based on the presentations. Also included is a list of attendees.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	
22a. NAME OF RESPONSIBLE INDIVIDUAL: Neil Gerr	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Computing Science and Statistics

Proceedings of the 20th Symposium on the Interface

Fairfax, Virginia, April 1988

Editors

EDWARD J. WEGMAN
DONALD T. GANTZ
JOHN J. MILLER

George Mason University, Fairfax, Virginia, U.S.A.

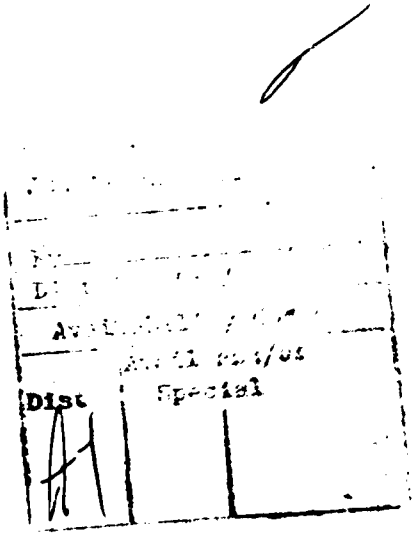
ASA

1988

AMERICAN STATISTICAL ASSOCIATION, ALEXANDRIA, VIRGINIA

The papers and discussions in this Proceedings volume are reproduced exactly as received from the authors. The authors have been encouraged to have their papers reviewed by a colleague prior to final preparation. These presentations are presumed to be essentially as given at the 20th Symposium on the Interface. This Proceedings volume is not copyrighted by the Association; hence, permission for reproduction must be obtained from the author, who holds the rights under the copyright law.

Authors in these Proceedings are encouraged to submit their papers to any journal of their choice. The ASA Board of Directors has ruled that publication in the Proceedings does not preclude publication elsewhere.



American Statistical Association
1429 Duke Street
Alexandria, Virginia 22314

PRINTED IN THE U.S.A.

Preface

The 20th Symposium on the Interface: Computing Science and Statistics was held on April 20th through 23rd, 1988. The 20th Symposium on the Interface was in a number of senses a watershed event for the Interface series. Begun in 1967 in Southern California as a one-day workshop meeting under the guidance of Arnie Goodman and Nancy Mann, it had matured over the years to a rather large scale event. The Board of Governors of the 18th Interface appointed an ad hoc committee to investigate incorporation of the Interface to preserve its financial and intellectual independence. At the 19th Interface a plan was presented by one of us (EJW) which included bylaws and a plan for incorporation. This plan was approved and in August of 1987, the Interface Foundation of North America, Inc. was formed as a non-profit, education corporation as the legal underpinning for the Interface Symposium series. The 20th Symposium was the first sponsored under the Interface Foundation banner. It was an auspicious start with a 50% increase in attendance, with the number of contributed papers nearly doubled, and with a healthy support from the federal funding agencies.

At the 18th Interface, much of the discussion in executive session focused on the direction of the meetings. The vision for the Interface Symposia obviously drew its focus from the interplay of computer science and statistics. While this was a largely unexplored area in 1967, the interface, in fact, has matured substantially and many of us thought that the interface was simply too broad and unfocused to remain the general theme of the Symposium. The 19th Interface Symposium, already well underway at that stage, was developed around the theme of *Large Scale Statistical Computing*. The 20th Interface Symposium was, in fact, developed from first blush with the theme of *Computationally Intensive Statistical Methods*. Much of factual detail about the 20th Symposium is contained in the front pages immediately following this Preface, e.g. lists of people involved, the past Interface Symposia, exhibitors, cooperating societies, the program schedule and the listing of papers in the technical program. We hope that these will be of interest.

We have, however, broken with past tradition which has organized the Proceedings around the format of the technical program. As with any Symposium, speakers often exhibit variances with the formal session titles in which they are scheduled to speak. In addition, when a theme is announced, it is often the case that contributed papers closely related to the themes of invited paper sessions are also submitted. We felt that, the Symposium now being history, it would be better to organize the Proceedings around the logical themes of the papers actually submitted for the Proceedings and making comparatively little distinction between what were invited papers and contributed papers. The clusters of papers are our choice and others may quibble with the classifications we made. Nonetheless, we hope that the organization of this volume makes logical sense to the reader and, more importantly, that the reader finds it to be useful.

Our major remaining task is to thank those people and organizations responsible for the success of the meeting. A major contributor to the success was our secretary, Jan P. Guenther. Many of the organization details that are attributed to the Program Chairman were in fact her ideas and we wish to publicly acknowledge our debt to her. A number of our graduate students, notably Masood Bolorforoush, Hung T. Le, Celesta Ball and Dale Penner spent long days in preparation and execution of many of the details. We also would like to acknowledge the patience of our families, notably the Wegman and the Guenther families, during the perparatory phases of the Symposium. The co-operating societies and organizations should be acknowledged as well. They are listed later in the program. Special note should be made of the Institute of Mathematical Statistics, the National Computer Graphics Association, the American Mathematical Society and the Society for Industrial and Applied Mathematics, each of which provided the Symposium organizers with free access to their membership lists. The National Bureau of Standards, now the National Institute of Standards and Engineering, printed the original announcement and mailed both the first and second sets of announcements.

The 20th Interface Symposium, as has been already mentioned, was the beneficiary of funding from several government agencies including the Air Force Office of Scientific Research under grant

number AFOSR-88-0154, the Army Research Office under grant number DAAL03-88-G-0020, the National Science Foundation under grant number DMS-8722898 and the Office of Naval Research under grant number N00014-88-J-1049. The editorial work of EJW on this volume was supported by the Air Force Office of Scientific Research under grant number AFOSR-87-0179, the Army Research Office under contract number DAAL03-87-K-0087, the National Science Foundation under grant number DMS-8701931 and the Virginia Center for Innovative Technology under contract number CIT/SPC-87-005. The latter contract also supported a portion of Jan Guenther's work.

Edward J. Wegman
Donald T. Gantz
John J. Miller
Fairfax, Virginia

TABLE OF CONTENTS

Symposium Chairman

Edward J. Wegman
Center for Computational Statistics
George Mason University
Fairfax, VA 22030
(703) 323-2723
EMAIL: EWEGMAN@GMUVAX (bitnet) or
EWEGMAN@GMUVAX.GMU.EDU (arpanet)

Symposium Coordinator and Exhibit Manager

Jan P. Guenther
Center for Computational Statistics
George Mason University
Fairfax, VA 22030
(703) 764-6170

Program Committee

David Allen
University of Kentucky

Chris Brown
University of Rochester

Martin Fischer
Defense Communication Engineering Center

Donald T. Gantz
George Mason University

Prem K. Goel
Ohio State University

Muhammed Habib
University of North Carolina

Mark E. Johnson
Los Alamos National Laboratory

Sallie Keller-McNulty
Kansas State University

Raoul LePage
Michigan State University

Don McClure
Brown University

John Miller
George Mason University

Mervin Muller
Ohio State University

Stephen Nash
George Mason University

Emanuel Parzen
Texas A and M University

Richard Ringeisen
Clemson University

Jerry Sacks
University of Illinois

David Scott
Rice University

Nozer Singpurwalla
George Washington University

Werner Stuetzle
University of Washington

Paul Tukey
Bell Communications Research

Past Interface Symposia

Southern California, 1967, 1968 1st and 2nd Symposia	Chair: Nancy Mann
Southern California, 1969 3rd Symposium	Chair: Ed Robinson
Southern California, 1971 4th Symposium	Chair: Mike Tarter Keynote Speakers: Richard Hamming and Frank Anscombe
Oklahoma State University, 1972 5th Symposium	Chair: Mitchell O. Locks Keynote Speaker: H. O. Hartley
University of California, Berkeley, 1973 6th Symposium	Chair: Michael Tarter Keynote Speaker: John Tukey
Iowa State University, 1974 7th Symposium	Chair: William J. Kennedy Keynote Speaker: Martin Wilk
University of California, Los Angeles, 1975 8th Symposium	Chair: James W. Frane Keynote Speaker: Edwin Kuh
Harvard University, 1976 9th Symposium	Chairs: David Hoaglin and Roy E. Welsch Keynote Speaker: John R. Rice
National Bureau of Standards, 1977 10th Symposium	Chair: David Hogben Keynote Speaker: Anthony Ralston
North Carolina State University, 1978 11th Symposium	Chairs: Ron Gallant and Thomas Gerig Keynote Speaker: Nancy Mann
University of Waterloo, 1979 12th Symposium	Chair: Jane F. Gentleman Keynote Speaker: D. R. Cox
Carnegie-Mellon University, 1981 13th Symposium	Chair: William F. Eddy Keynote Speaker: Brad Efron
Rensselaer Polytechnic Institute, 1982 14th Symposium	Chairs: John W. Wilkinson, Karl W. Heiner and Richard Sacher Keynote Speaker: John Tukey
IMSL, Inc (held in Houston), 1983 15th Symposium	Chair: James Gentle Keynote Speaker: Richard Hamming
University of Georgia (held in Atlanta), 1984 16th Symposium	Chair: Lynne Billard Keynote Speaker: George Marsalgia
University of Kentucky, 1985 17th Symposium	Chair: David Allen Keynote Speaker: John C. Nash

Past Interface Symposia (Continued)

Colorado State University, 1986 18th Symposium	Chair: Thomas Boardman Keynote Speaker: John Tukey
Temple University (held in Phi'adelphia), 1987 19th Symposium	Chair: Richard Heiberger Keynote Speaker: Gene Golub
George Mason University, 1988 20th Symposium	Chair: Edward J. Wegman Keynote Speaker: Brad Efron

Future Interface Symposia

University of South Florida, 1989 21st Symposium	Chairs: Ken Berk and Linda Malone
Michigan State University, 1990 22nd Symposium	Chair: Raoul LePage

General Information

The 20th Symposium represents a milestone in the development of the interface between computing science and statistics. In August, 1987 the Interface Foundation of North America was incorporated as a non-profit, educational corporation whose main charter is to provide the legal entity underpinning the Symposium series. The Foundation represents a maturation of the Symposium series and ensures its continuation as an independent meeting focused on the interface. The 20th Symposium is the first held under the auspices of the Foundation.

Theme: – Computationally Intensive Statistical Methods

Keynote Address: – "Computationally intensive statistical inference"
Bradley Efron, Department of Statistics, Stanford University

Invited Papers: – There are 60 invited papers including several with invited discussion organized into 23 sessions. In addition to the plenary session with the keynote address by Brad Efron, there are three special invited lectures featuring Jerome Friedman, George E. P. Box and Thomas Banchoff.

Contributed Papers: – There are 128 contributed papers scheduled in 26 sessions.

Exhibitors

Ametek Computer Corporation
606 East Huntington Drive
Monrovia, CA 91016
(714) 599-4662

Automatic Forecasting Systems, Inc.
P. O. Box 563
Hatboro, PA 19040
(215) 675-0652

North Holland/Elsevier Publishers
P. O. Box 1991
1000 BZ Amsterdam
The Netherlands

Numerical Algorithms Group
1101 31st Street, Suite 100
Downers Grove, IL 60515
(312) 971-2337

BBN Software
10 Fawcett Street
Cambridge, MA 02238
(617) 873-8116

BMDP Statistical Software, Inc.
1440 Sepulveda Boulevard, Suite 316
Los Angeles, CA 90025
(213) 479-7799

Intel Scientific Computers
15201 NW Greenbrier Parkway
Beaverton, OR 97006
(503) 629-7631

Marcel-Dekker, Inc.
270 Madison Avenue
New York, NY 10016
(212) 696-9000

IMSL, Inc.
2500 ParkWest Tower One
2500 CityWest Boulevard
Houston, TX 77042-3020
(713) 782-6060

Springer-Verlag, Inc.
175 Fifth Avenue
New York, NY 10010
(212) 460-1600

SYSTAT, Inc.
1800 Sherman Avenue
Evanston, IL 60201
(312) 864-5670

TCI Software
1190 Foster Road
Las Cruces, NM 88001
(505) 522-4600

Tektronix, Inc.
M.S. 48-300, Industrial Park
Beaverton, OR 97077
(503) 627-7111

Wadsworth & Brooks/Cole
Advanced Books and Software
10 Davis Drive
Belmont, CA 94002
(415) 595-2350

Short Course

Forecasting on the IBM-PC - A Survey, Wednesday, April 20, 9:00 a.m. to 4:30 p.m., David P. Reilly, Automatic Forecasting Systems, Inc., P. O. Box 563, Hatboro, PA 19040, (215) 675-0652

Cooperating Societies

American Mathematical Society
P. O. Box 6248
Providence, RI 02940

American Statistical Association
1429 Duke Street
Alexandria, VA 22314

International Association for Statistical Computing
NTDH
P. O. Box 145
N-7701 Steinkjer
Norway

Institute of Mathematical Statistics
3401 Investment Boulevard, Suite 7
Hayward, CA 94545

National Computer Graphics Association
2722 Merilee, Suite 200
Fairfax, VA 22031

Operations Research Society of America
Mount Royal and Guilford Avenues
Baltimore, MD 21202

Society for Industrial and Applied Mathematics
1400 Architects Building
117 South 17th Street
Philadelphia, PA 19103

Virginia Academy of Science Chapter of ASA
c/o Golde I. Holtzman
Department of Statistics
Virginia Tech
Blacksburg, VA 24061

Washington Statistical Society
P. O. Box 70843
Washington, DC 20024-0843

Program Schedule

Date and Time	Session Title
Thursday, April 21	
8:45 a.m. - 9:45 a.m.	Keynote Address: Computationally Intensive Statistical Inference
10:00 a.m. - 12:00 noon	Computational Aspects of Time Series Analysis Inference and Artificial Intelligence Computational Discrete Mathematics Contributed: Software Tools Contributed: Image Processing I Contributed: Bootstapping and Related Computational Methods
1:30 p.m. - 3:30 p.m.	Special Invited Lecture I Image Processing and Spatial Processes Parallel Computing Architectures Contributed: Statistical Methods I Contributed: Hardware and Software Reliability Contributed: Applications I
3:45 p.m. - 5:45 p.m.	Special Invited Session for Recent Ph.D.'s Simulation Symbolic Computation and Statistics Contributed: Statistical Graphics Contributed: Models of Imprecision in Expert Systems Contributed: Time Series Methods
Friday, April 22	
8:00 a.m. - 10:00 p.m.	Computer-Communication Networks Supercomputing, Design of Experiments and Bayesian Analysis, Part 1 Numerical Methods in Statistics Contributed: Probability and Stochastic Processes Contributed: Statistical Methods II Contributed: Nonparametric and Robust Techniques
10:15 a.m. - 12:15 p.m.	Special Invited Lecture II Supercomputing, Design of Experiments and Bayesian Analysis, Part 2 Neural Networks Contributed: Applications II Contributed: Image Processing II Contributed: Simulation I

2:00 p.m. - 4:00 p.m.	Tales of the Unexpected: Successful Interdisciplinary Research
	Density Estimation and Smoothing
	Object Oriented Programming
	Contributed: Numerical Methods
	Contributed: Bayesian Methods
	Contributed: Expert Systems in Statistics

Saturday, April 23

8:30 a.m. - 10:30 a.m.	Computational Aspects of Simulated Annealing
	Dynamical High Interaction Graphics
	Contributed: Statistical Methods III
	Contributed: Simulation II
	Contributed: Biostatistics Applications
	Contributed: Discrete Mathematical Methods
10:45 a.m. - 12:45 p.m.	Special Invited Lecture III
	Entropy Methods
	Contributed: Information Systems, Databases and Statistics
	Contributed: Parallel Computing
	Contributed: Density and Function Estimation
	Contributed: Statistical Methods IV

Technical Program

WEDNESDAY, APRIL 20, 1988

9:00 a.m. - 4:30 p.m.

Short Course - Forecasting on the IBM-PC, David Reilly, Automatic Forecasting Systems, Inc.

THURSDAY, APRIL 21, 1988

8:45 a.m. - 9:45 a.m.

Plenary Session, Chaired by: Edward J. Wegman, George Mason University

"Computationally intensive statistical inference," Bradley Efron, Stanford University

10:00 a.m. - 12:00 noon

Computational Aspects of Time Series Analysis, Chaired by: Emanuel Parzen, Texas A & M University

"Recent progress in algorithms and architectures for time series analysis," George Cybenko, Tufts University

"Numerical approach to non-gaussian smoothing and its application," Genshiro Kitagawa, The Institute of Statistical Mathematics

Discussants - Will Gersch, University of Hawaii and H. Joseph Newton, Texas A & M University

THURSDAY, APRIL 21, 1988

10:00 a.m. - 12:00 noon
Inference and Artificial Intelligence, Chaired by: N. Singpurwalla, George Washington University

"Spectral Analysis on a LISP machine," Don Percival, University of Washington

"DeFinetti's approach to group decision making," Richard Barlow, University of California, Berkeley

"Meta-analysis," Ingram Olkin, Stanford University

10:00 a.m. - 12:00 noon
Computational Discrete Mathematics, Chaired by: Rich Ringeisen, Clemson University

"Discrete structures and reliability computations," James P. Jarvis, Clemson University and Douglas R. Shier, College of William and Mary

"Random graphs," Edward R. Scheinerman, The Johns Hopkins University

"Structure and finiteness conditions on graphs," Neil Robertson, Ohio State University

10:00 a.m. - 12:00 noon
Contributed Papers: Software Tools, Chaired by: Leonard Hearne, George Mason University

"An introduction to CART^{tm}: classification and regression trees," Gerard T. LaVarnway, Norwich University

"Noise appreciation: analyzing residuals using RS/Explore," David A. Burn and Fanny O'Brien, BBN Software Products Corporation

"COSTAR: an environment for computer-guided data analysis," David A. Whitney and Ilya Schiller, TASC

"A closer look at symbolic computation," William M. Makuch, General Electric Corporation and John W. Wilkinson, Rensselaer Polytechnic Institute

10:00 a.m. - 12:00 noon
Contributed Papers: Image Processing I, Chaired by: A. K. Sood, George Mason University

"Image analysis of a turbulent object using fractal parameters," Amar Ait-Kheddache, North Carolina State University

"Identification of closed figures," Jeff Banfield, Montana State University and Adrian Raftery, University of Washington

"Compression of image data using arithmetic coding," Ahmed H. Desoky and Thomas Klein, University of Louisville

"Image analysis of the microvascular system in the rat cremaster muscle," C. O'Connor, P. D. Harris, A. Desoky and G. Ighodaro, University of Louisville

"Automatic detection of the optic nerve in color images of the retina," Norman Katz, Subhasis Chaudhuri, and Michael Goldbaum, University of California, San Diego and Mark Nelson, Radford Company

10:00 a.m. - 12:00 noon

Contributed Papers: Bootstrapping and Related Computational Methods, Chaired by: Richard Bolstein, George Mason University

"A Monte Carlo study of cross-validation and the C_p criterion for model selection in multiple linear regression," Robert M. Boudreau, Virginia Commonwealth University

"Bootstrapping regression strategies," David Brownstone, University of California, Irvine

"Bootstrapping the missed regression model with reference to the capital and energy complementarity debate," Baldev Raj, Wilfred Laurier University

"Efficient data sensitivity computation for maximum likelihood estimation," Daniel Chin and James C. Spall, The Johns Hopkins University

"Bootstrap procedures in random effect models for comparing response rates in multi-center clinical trials," Michael F. Miller, Hoechst-Roussel Pharmaceuticals, Inc.

1:30 p.m. - 2:45 p.m.

Special Invited Lecture I, Chaired by: Jim Filliben, National Bureau of Standards

"Fitting functions to scattered noisy data in high dimensions," Jerome Friedman, Stanford University

1:30 p.m. - 3:30 p.m.

Image Processing and Spatial Processes, Chaired by: Don McClure, Brown University

Introduction, Don McClure, Brown University

"A multilevel-multiresolution technique for image analysis and robot vision via renormalization group ideas," Basilis Gidas, Brown University

"A mathematical approach to expert system construction," Alan Lippman, Brown University

THURSDAY, APRIL 21, 1988

1:30 p.m. - 3:30 p.m.
Parallel Computing Architectures, Chaired by: Chris Brown, University of Rochester

"Experiences with the BBN Butterfly^{tm} parallel processor," John Mellor-Crummy, University of Rochester

"Statistical computing on a hypercube," George Ostrouchov, Oak Ridge National Lab

"Asychronous iteration," William F. Eddy and Mark Schervish, Carnegie-Mellon University

1:30 p.m. - 3:30 p.m.
Contributed Papers: Statistical Methods I, Chaired by: Walter Liggett, National Bureau of Standards

"An example of the use of a Bayesian interpretation of multiple discriminant analysis results," James R. Nolan, Siena College

"Real-time classification and discrimination among components of a mixture distribution," Douglas A. Samuelson, International Telesystems Corporation

"Comparison of three 'local model' classification methods," Daniel Normolle, University of Michigan

"Application of posterior approximation techniques for the ordered Dirichlet distribution," Thomas A. Mazzuchi and Refik Soyer, George Washington University

"Unbiased estimates of multivariate general moment functions of the population and application to sampling without replacement for a finite population," Nabih N. Mikhail, Liberty University

1:30 p.m. - 3:30 p.m.
Contributed Papers: Hardware and Software Reliability, Chaired by: Asit Basu, University of Missouri

"Linear prediction of failure times of a repairable system," M. Ahsanullah, Rider College

"The simulation of life tests with random censoring," Joseph C. Hudson, GMI Engineering and Management Institute

"The use of general modified exponential curves in software reliability modeling," Taghi M. Khoshgoftaar, Florida Atlantic University

"A model for information censoring," William A. Link, Patuxent Wildlife Research Center

"Increasing reliability of multiversion fault-tolerant software design by modulation," Junryo Miyashita, California State University, San Bernardino

THURSDAY, APRIL 21, 1988

1:30 p.m. - 3:30 p.m.

Contributed Papers: Applications I, Chaired by: Susannah Schiller, National Bureau of Standards

"Classifying linear mixtures with an application to high resolution gas chromatography," William S. Rayens, University of Kentucky

"Bias of animal trend estimates," Paul H. Geissler and William A. Link, Patuxent Wildlife Research Center

"A non-random walk through futures prices of the British pound," William S. Mallios, California State University, Fresno

"A stochastic extension of Petri net graph theory," L. M. Anneberg, Wayne State University

"Neural Petri nets," N. H. Chamas, Wayne State University

3:45 p.m. - 5:45 p.m.

Special Invited Session for Recent Ph.D.'s, Chaired by: John J. Miller, George Mason University

"Additive principal components: a method for estimating equations with small variance from multivariate data," Deborah Donnell, Bellcore

"Gamma processes, paired comparisons and ranking," Hal Stern, Harvard University

"Smoothing data with correlated errors," Naomi Altman, Cornell University

"The data viewer: program for graphical data analysis," Catherine Hurley, University of Waterloo

3:45 p.m. - 5:45 p.m.

Simulation, Chaired by: Donald T. Gantz, George Mason University

"Random variables for supercomputers," George Marsaglia, Florida State University

"Computational statistics in experimental design for studies of variability," John Ramberg, University of Arizona

"Linear combinations of estimators of the variance of the sample mean," Bruce W. Schmeiser, Purdue University

THURSDAY, APRIL 21, 1988

3:45 p.m. - 5:45 p.m.

Symbolic Computation and Statistics, Chaired by: William S. Rayens, University of Kentucky

"Some applications of symbol manipulation in statistical analysis," Kathryn M. Chaloner, University of Minnesota

"Symbolic computation in statistical decision theory," Marietta Tretter, Texas A & M University

"Partial differentiation by computer with applications to statistics," John W. Sawyer, Jr., Texas Tech University

3:45 p.m. - 5:45 p.m.

Contributed Papers: Statistical Graphics, Chaired by: Robert Launer, Army Research Office

"Visual multidimensional geometry with applications," Alfred Inselberg, IBM Scientific Center, Los Angeles and Bernard Dimsdale, University of California

"Some graphical representations of multivariate data," Masood Bolorforoush and Edward J. Wegman, George Mason University

"Graphical representations of main effects and interaction effects in a polynomial regression on several predictors," William DuMouchel, BBN Software Products Corporation

"Chernoff faces: a PC implementation," Mohammad Dadashzadeh, University of Detroit

3:45 p.m. - 5:45 p.m.

Contributed Papers: Models of Imprecision in Expert Systems, Chaired by: Mark Youngren, George Washington University

"Fusion and propagation of graphical belief models," Russell Almond, Harvard University

"Belief function computations for paired comparisons," David Tritchler and Gina Lockwood, Ontario Cancer Institute

"Variants of Tierney-Kadane," Guenter Weiss and H. A. Howlader, University of Winnepeg

"Dynamically updating relevance judgements in probabilistic information systems via users' feedback," Peter Lenk and Barry D. Floyd, New York University

"Computational requirements for inference methods in expert systems: a comparative study," Ambrose Goicoechea, George Mason University

THURSDAY, APRIL 21, 1988

3:45 p.m. - 5:45 p.m.

Contributed Papers: Time Series Methods, Chaired by: Neil Gerr, Office of Naval Research

"Inference techniques for a class of exponential time series," V. Chandrasekar and Peter Brockwell, Colorado State University

"Some recursive methods in time series analysis," Q. P. Duong, Bell Canada

"Time series in a microcomputer environment," John Henstridge, Numerical Algorithms Group

"Smoothing irregular time series," Keith W. Hipel, University of Waterloo, A. I. McLeod, The University of Western Ontario and Byron Bodo, Ministry of the Environment

"Computation of the theoretical autocovariance function of multivariate ARMA processes," Stefan Mittnik, SUNY at Stony Brook

FRIDAY, APRIL 22, 1988

8:00 a.m. - 10:00 a.m.

Computer-Communication Networks, Chaired by: Martin Fischer, Defense Communication Engineering Center

"Introduction to packet switching networks," Jeffrey Mayersohn, BBN Communication Corporation

"Electronic mail - a valuable augmentation tool for scientists," Elizabeth Feinler, SRI International

"Networks to support science," Stephen Wolff, National Science Foundation

FRIDAY, APRIL 22, 1988

8:00 a.m. - 10:00 a.m.

Supercomputing, Design of Experiments and Bayesian Analysis, Part I, Chaired by: Jerry Sacks, University of Illinois

"Acceleration methods for Monte Carlo integration by Bayesian inference," John Geweke, Duke University

"Software for Bayesian analysis: current status and additional needs," Prem K. Goel, Ohio State University

"Some numerical and graphical stategies for implementing Bayesian methods," Adrian Smith, University of Nottingham

8:00 a.m. - 10:00 a.m.

Numerical Methods for Statistics, Chaired by: Stephen Nash, George Mason University

"Interior point methods for linear programming," Paul Boggs, National Bureau of Standards

"Block iterative methods for parallel optimization," Stephen Nash and Ariela Sofer, George Mason University

"New methods for B-differentiable functions: theory and applications," Jong-Shi Pang, The Johns Hopkins University

8:00 a.m. - 10:00 a.m.

Contributed Papers: Probability and Stochastic Processes, Chaired by: Yash Mittal, National Science Foundation

"Moving window detection for 0-1 Markov trials," Joseph Glaz, University of Connecticut, Philip C. Hormel, CIBA-GEIGY Corporation and Bruce McK. Johnson, University of Connecticut

"Maximum queue size and hashing with lazy deletion," Claire M. Mathieu, Laboratoire d'Informatique de l'Ecole Normale Superieure and Jeffrey S. Vitter, Brown University

"On the probability integrals of the multivariate normal," Dror Rom and Sanat Sarkar, Temple University

"Computational aspects of harmonic signal detection," Keh-Shin Lii and Tai-Houn Tsou, University of California, Riverside

"Maximum likelihood estimation of discrete control processes: theory and application," John Rust, University of Wisconsin

FRIDAY, APRIL 22, 1988

8:00 a.m. - 10:00 a.m.

Contributed Papers: Statistical Methods II, Chaired by: Cliff Sutton, George Mason University

"Computing extended maximum likelihood estimates in generalized linear models," Douglas B. Clarkson, IMSL, Inc. and Robert I. Jennrich, University of California, Los Angeles

"Assessment of prediction procedures in multiple regression analysis," Victor Kipnis, University of Southern Florida

"Estimation of the variance matrix for maximum likelihood parameters by quasi-Newton methods," Linda Pickle, National Cancer Institute and Garth P. McCormick, George Washington University

"Variable selection in multivariate multiresponse permutation procedures," Eric P. Smith, Virginia Tech

"The effect of small covariate-criterion correlations on analysis of covariance," Michael J. Rovine, A. von Eye and P. Wood, Pennsylvania State University

8:00 a.m. - 10:00 a.m.

Contributed Papers: Nonparametric and Robust Techniques, Chaired by: Paul Speckman, University of Missouri

"Robustness of weighted estimators of location: a small sample survey," Greg Campbell and Richard I. Shrager, NIH

"A comparison of Spearman's footrule and rank correlation coefficient with exact tables and approximations," LeRoy A. Franklin, Indiana State University

"Approximations of the Wilcoxon test in small samples with lots of ties," Arthur R. Silverberg, Food and Drug Administration

"Simulated power comparisons of MRPP rank tests and some standard score tests," Derrick S. Tracy and Khushnood A. Khan, University of Windsor

10:15 a.m. - 12:15 p.m.

Special Invited Lecture II, Chaired by: Mervin Muller, Ohio State University

"Some modern quality improvement techniques and their computing implications," George E. P. Box, University of Wisconsin

Special invited discussion, Gerald J. Hahn, GE CRD and Gregory B. Hudak, Scientific Computing Associates

FRIDAY, APRIL 21, 1988

10:15 a.m. - 12:15 p.m.
Supercomputing, Design of Experiments and Bayesian Analysis, Part II, Chaired by: Prem K. Goel, Ohio State University

"Supercomputer-aided design," Jerry Sacks, University of Illinois

"A Bayesian approach to the design and analysis of computer experiments," Toby Mitchell, Oak Ridge National Lab

10:15 a.m. - 12:15 p.m.
Neural Networks, Chaired by: Muhammed Habib, University of North Carolina

"Statistical learning networks: a unifying view," Andrew R. Barron, University of Illinois and Roger L. Barron, Barron Associates, Inc.

"Stochastic models of neuronal behavior," Gopinath Kallianpur, University of North Carolina

"Inference for stochastic models for neural networks," Muhammed Habib, University of North Carolina and A. Thavaneswaran, Temple University

10:15 a.m. - 12:15 p.m.
Contributed Papers: Applications II, Chaired by: Brian Woodruff, Air Force Office of Scientific Research

"Space Balls! or estimating diameter distributions of polystyrene microspheres," Susannah Schiller and Charles Hagwood, National Bureau of Standards

"Comparing sample reuse methods at FHA - an empirical approach," Thomas N. Herzog, U. S. Department of Housing and Urban Development

"Maximum entropy and its application to linguistic diversity," R. K. Jain, Memorial University of Newfoundland

"Encoding and processing of Chinese language - a statistical structural approach," Chaiho C. Wang, George Washington University

"The elimination of quantization bias using dither," Martin J. Garbo and Douglas M. Dreher, Hughes Aircraft Company

10:15 a.m. - 12:15 p.m.
Contributed Papers: Image Processing II, Chaired by: Refik Soyer, George Washington University

"Maximum entropy and the nearly black image," Iain Johnstone, Stanford University and David Donoho, University of California, Berkeley

"A probabilistic approach to range image description," Arun Sood, George Mason University and E. Al-Hujazi, Wayne State University

"An empirical Bayes decision rule of two-class pattern recognition for one-dimensional parametric distributions," Tze Fen Li, Rutgers University

"Statistical modeling of a priori information for image processing problems." Z. Liang, Duke University Medical Center

"Advanced statistical computations improve image processing applications, Bobby Saffari, Generex Corporation

10:15 a.m. - 12:15 p.m.

Contributed Papers: Simulation I, Chaired by: Bill DuMouchel, BBN

"On comparative accuracy of multivariate nonnormal random number generators," Lynne K. Edwards, University of Minnesota

"Bayesian analysis using Monte Carlo integration: an effective methodology for handling some difficult problems in statistical analysis," Leland Stewart, Lockheed Research Laboratory

"A squeeze method for generating exponential power variates," Dean M. Young, Baylor University

"Mixture experiments and fractional factorials used to tailor large-scale computer simulation," T.K. Gardenier, TKG Consultants, Ltd.

"Simulating stationary Gaussian ARMA time series," Terry J. Woodfield, SAS Institute, Inc.

2:00 p.m. - 4:00 p.m.

Tales of the Unexpected: Successful Interdisciplinary Research, Chaired by: Sallie McNulty, Kansas State University

"Some statistical problems in meteorology," Grace Wahba, University of Wisconsin

"Modeling parallelism, an interdisciplinary approach," Elizabeth Unger, Kansas State University

"Mice, rain forests and finches: experiences collaborating with biologists," Douglas Nychka, North Carolina State University

Discussion: Jerome Sacks, University of Illinois

FRIDAY, APRIL 22, 1988

2:00 p.m. - 4:00 p.m.
Density Estimation and Smoothing, Chaired by: David Scott, Rice University

"XploRe: computing environment for exploratory regression and density estimation methods," Wolfgang Härdle, University of Bonn

"Curve estimation with applications to mapping and risk decomposition," Michael Tarter, University of California, Berkeley

"Interactive multivariate density estimation in the S package," David Scott, Rice University

2:00 p.m. - 4:00 p.m.
Object Oriented Programming, Chaired by: Werner Stuetzle, University of Washington

"Object oriented programming: a tutorial," Wayne Oldford, University of Waterloo

"An object oriented toolkit for plotting and interface construction," Robert Young, Schlumburger, Palo Alto Research Center

"An outline of Arizona," John MacDonald, University of Washington

2:00 p.m. - 4:00 p.m.
Contributed Papers: Numerical Methods, Chaired by: Ariela Sofer, George Mason University

"A theorgy of quadrature in applied probability: a fast algorithmic approach," Allen Don, Long Island University

"Higher order functions in numerical programming," David Gladstein, ICAD

"A numerical comparison of EM and quasi-Newton type algorithms for finding MLE's for a mixture of normal distributions," Richard J. Hathaway, John W. Davenport and Margaret Anne Pierce, Georgia Southern College

"Numerical algorithms for exact calculations of early stopping probabilities in one-sample clinical trials with censored exponential responses," Brenda MacGibbon, Concordia University, Susan Groshen, University of Southern California and Jean-Guy Levreault, University of Montreal

"An application of quasi-Newton methods in parametric empirical Bayes calculations," David Scott, Concordia University

FRIDAY, APRIL 22, 1988

2:00 p.m. - 4:00 p.m.
Contributed Papers: Bayesian Methods, Chaired by: William F. Eddy, Carnegie-Mellon University

"Approaches for empirical Bayes confidence intervals with application to exponential scale parameters," Alan E. Gelfand and Bradley P. Carlin, University of Connecticut

"A data analysis and Bayesian framework for errors-in-variables," John H. Herbert, Department of Energy

"Bayesian diagnostics for almost any model," Robert E. Weiss, University of Minnesota

"An iterative Bayes method for classifying multivariate observations," Duane E. Wolting, Acrojet Tech Systems Company

"A Bayesian model of information conbination from noisy sensors," G. Anandalingam, University of Pennsylvania

2:00 p.m. - 4:00 p.m.
Contributed Papers: Expert Systems in Statistics: Chaired by Khalid Abouri, George Washington University

"Inside a statistical expert system: implementation of the ESTES expert system," Paula Hietala, University of Tampere, Finland

"Knowledge-based project management: work effort estimation," Vijay Kanabar, University of Winnipeg

"Combining knowledge acquisition and classical statistical techniques in the development of a veterinary medical expert system," Mary McLeish, University of Guelph

"The effect of measurement error in a machine learning system," David L. Rumpf and Mieczyslaw M. Kokar, Northeastern University

"An expert system for prescribing statistical tests of non-parametric and simple parametric designs," Gary Tubb, University of South Florida

SATURDAY, APRIL 23, 1988

8:30 a..m. - 10:30 a.m.

Computational Aspects of Simulated Annealing, Chaired by: Mark E. Johnson, Los Alamos National Lab

"Computational experience with simulated annealing," Daniel G. Brooks and William A. Verdini, Arizona State University

"Simulated annealing in optimal design construction," Ruth K. Meyer, St. Cloud State University and Christopher J. Nachtsheim, University of Minnesota

"A simulated annealing approach to mapping DNA," Larry Goldstein and Michael J. Waterman, University of Southern California

8:30 a.m. - 10:30 a.m.

Dynamical High Interaction Graphics, Chaired by: Paul Tukey, Bellcore

"Determining properties of minimal spanning trees by local sampling," Allen McIntosh, Bellcore and William Eddy, Carnegie-Mellon University

"Data animation," Rick Becker, AT&T Bell Labs and Paul Tukey, Bellcore

"Dimensionality constraints on projection and section views of higher dimensional loci," George Furnas, Bellcore

8:30a.m. - 10:30 a.m

Contributed Papers: Statistical Methods III, Chaired by: Thomas Mazzuchi, George Washington University

"Simultaneous confidence intervals in the general linear model," Jason C. Hsu, Ohio State University

"Empirical likelihood ratio confidence regions," Art Owen, Stanford University

"An approximate confidence interval for the optimal number of mammography x-ray units in the Dallas-Fort Worth metropolitan area," Roger W. Peck, University of Rhode Island

"Optimizing linear functions of random variables having a joint multinomial or multivariate normal distribution," Josephina P. de los Reyes, University of Akron

"On covariances of marginally adjusted data," James S. Weber, Roosevelt University

8:30 a.m. - 10:30 a.m.

Contributed Papers: Simulation II, Chaired by : Robert Jernigan, American University

"SIMDAT and SIMEST: differences and convergences," James R. Thompson, Rice University

"Simulation and stochastic modeling for the spatial allocation of multi-categorical resources," Richard S. Segall, University of Lowell

"Robustness study of some random variate generators," Lih-Yuan Deng, Memphis State University

"Testing multiprocessing random number generators," Mark J. Durst, Lawrence Livermore National Laboratory

"An approach for generations of two variable sets with a specified correlation and first and second sample moments," Mark Eakin and Henry D. Crockett, University of Texas at Arlington

8:30 a.m. - 10:30 a.m.
Contributed Papers: Biostatistics Applications, Chaired by: Nancy Flournoy, National Science Foundation

"An algorithm to identify changes in hormone patterns," Morton B. Brown, Fred J. Karsch and Benoit Malpaux, University of Michigan

"Applying microcomputer techniques to multiple cause of death data: from magnetic tape to artificial intelligence," Giles Crane, New Jersey State Department of Health

"Spline estimation of death density using census and vital statistics data," John J. Hsieh, University of Toronto

"Optimum experimental design for sequential clinical trials," Richard Simon, National Cancer Institute

"Bayes estimation of cerebral metabolic rate of glucose in stroke patients," P. David Wilson, University of South Florida, S. C. Huang and R. A. Hawkins, UCLA School of Medicine

8:30 a.m. - 10:30 a.m.
Contributed Papers: Discrete Mathematical Methods, Chaired by: Donald Gantz, George Mason University

"Minimum cost path planning in the random traversability space," A. Meystel, Drexel University

"Algorithms to reconstruct a convex set from sample points," Marc Moore, Ecole Polytechnique Montreal and McGill University, Y. Lemay, Bell Canada, and S. Archambault, Ecole Polytechnique Montreal

"On the geometric probability of discrete lines and circular arcs approximating arbitrary object boundaries," Chang Y. Choo, Worchester Polytechnic Institute

"Application of orthogonalization procedures to fitting tree-structured models," Cynthia O. Siu, The Johns Hopkins University

"Evaluation of functions over lattices," Michael Conlon, University of Florida

SATURDAY, APRIL 23, 1988

10:45 a.m. - 12:00 noon

Special Invited Lecture III, Chaired by: Sally Howe, National Bureau of Standards

"Visualizing high dimensional spaces," Thomas Banchoff, Brown University

10:45 a.m. - 12:45 p.m.

Entropy Methods, Chaired by: Raoul LePage, Michigan State University

"Introduction to relative entropy methods," John Shore, Entropic Processing Corporation

"Structural covariance matrices and 2-dimensional spectra," John Burg, Entropic Processing Corporation

"Matrix completion and determinants," Charlie Johnson, College of William and Mary

10:45 a.m. - 12:45 p.m.

Contributed Papers: Information Systems, Databases and Statistics, Chaired by: Robert Teitel, Teitel Data Services

"Information systems and statistics," Nancy Flournoy, National Science Foundation

"Is there a need for a statistical knowledge base?" Z. Chen, Louisiana State University

"An alternate methodology for subject database planning," Craig W. Slinkman, Henry D. Crockett, and Mark Eakin, University of Texas at Arlington

"A sensitivity analysis of the Herfindal-Hirschman Index," James R. Knaub, Jr., U. S. Department of Energy

"Statistical methods for document retrieval and browsing," Jan Pedersen, Xerox PARC and John Tukey and P. K. Halvorsen

10:45 a.m. - 12:45 p.m.

Contributed Papers: Parallel Computing, Chaired by: Joseph Brandenburg, INTEL Scientific Computers

"Programming the BBN butterfly parallel processor," Pierre duPont, BBN Advanced Computers

"A tool to generate parallel FORTRAN code for the Intel iPSC/2 hypercube," Carlos Gonzalez, J. Chen and J. Sarma, George Mason University

"All-subsets regression on a hypercube multiprocessor," Peter Wollan, Michigan Technological University

"Multiply twisted N-cubes for multiprocessor parallel computers," T.H. Shiau, University of Missouri, Columbia

"Markov chains arising in collective computation networks with additive noise," R.H. Baran, Naval Surface Warfare Center

SATURDAY, APRIL 23, 1988

10:45 a.m. - 12:45 p.m.

Contributed Papers: Density and Function Estimation, Chaired by: Celesta Ball, George Mason University

"The L_1 asymptotically optimal kernel estimate," Luc Devroye, McGill University

"Derivative estimation by polynomial-trigonometric regression," Paul Speckman, University of Missouri, Columbia and R.L. Eubank, Southern Methodist University

"A pooled error density estimate for the bootstrap," Walter Liggett, National Bureau of Standards

"Efficient algorithms for smoothing spline estimation of functions with or without discontinuities," Jyh-Jen Horng Shiau, University of Missouri, Columbia

"On the convergence of variable bandwidth kernel estimators of a density function," Ting Yang, University of Cincinnati

10:45 a.m. - 12:45 p.m.

Contributed Papers: Statistical Methods IV, Chaired by: LeRoy A. Franklin, Indiana State University

"Stochastic test statistics," P. Warwick Millar, University of California, Berkeley

"It's time to stop!," Hubert Lilliefors, George Washington University

"The effects of heavy tailed distributions on the two sided k-sample Smirnov test," Henry D. Crockett and M. M. Whiteside, University of Texas at Arlington

"Performance of several one sample procedures," David Turner, Utah State University

"Exact power calculation for the chi-square test of two proportions," Carl E. Pierchala, Food and Drug Administration

AVAILABILITY OF PROCEEDINGS

18, 19, 20th (1986, 87, 88)	American Statistical Association 1429 Duke Street Alexandria, Virginia 22314-3402
15, 16, 17th (1983, 84, 85)	North Holland Publishing Company Amsterdam The Netherlands
13, 14th (1981, 82)	Springer-Verlag New York, Inc. 175 Fifth Avenue New York, New York 10010
12th (1979)	Jane F. Gentleman Statistics Canada Social & Economic Statistics Division Coats Building, 24th Floor Tunney's Pasture Ottawa, Ontario Canada K1A 0T6
11th (1978)	Institute of Statistics North Carolina State University P.O. Box 5457 Raleigh, North Carolina 27650
10th (1977)	Statistical Engineering Laboratory Applied Mathematics Division National Bureau of Standards U.S. Department of Commerce Washington, DC 20234
9th (1976)	American Statistical Association 1429 Duke Street Alexandria, Virginia 22314-3402
8th (1975)	Health Sciences Computing Facility, AV-111 Center for Health Sciences University of California Los Angeles, California 90024
7th (1974)	Statistical Numerical Analysis and Data Processing Section 117 Snedecor Hall Iowa State University Ames, Iowa 50010
4, 5, 6th (1971, 72, 73)	Western Periodicals Company 13000 Raymer Street North Hollywood, California 91605

I. KEYNOTE ADDRESS

Computer Intensive Statistical Inference

Bradley Efron, Stanford University

Computer Intensive Statistical Inference

Bradley Efron
Department of Statistics
Stanford University

Abstract: We discuss three recent data analyses which illustrate making statistical inferences (finding significance levels, confidence intervals and standard errors) with the critical assistance of the computer. The first example concerns a permutation test for a linear model situation with several covariates. We provide a computer-based compromise between complete randomization and optimum design, partially answering the question "how much randomization is enough?" A problem in particle physics provides the second example. We use the bootstrap to find a good estimator for an interesting decay probability and then to obtain a believable confidence interval. The third problem involves a long-running cancer trial in which the z-value in favor of the more rigorous treatment wandered extensively during the course of the experiment. A dubious theory, which suggests that the wandering is just due to random noise, is rendered more believable by a bootstrap analysis. All three examples illustrate the tendancy for computer-based inference to raise new points in statistical theory. [Editors note: Professor Efron provided this abstract together with the following examples which were his handout to summarize his Keynote Address.]

Mouse Data: Ordinary Permutation Test

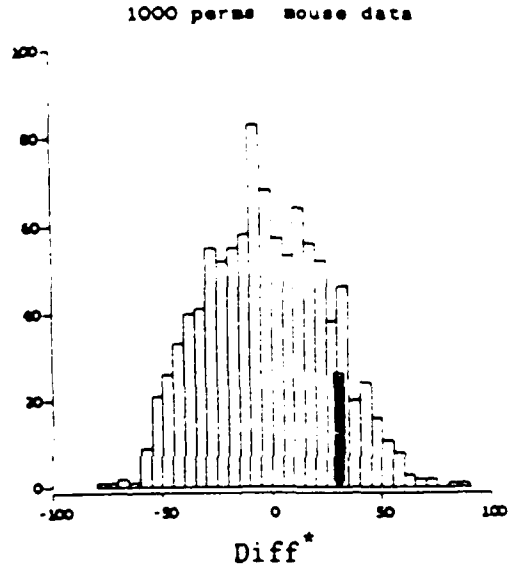
* Two groups of mice, "A"=Treatment (7 mice) and "B"=Control (9 mice).

* For each mouse, measured survival time in days after surgery

A: 94 197 16 38 99 141 23 Mean= 86.9
B: 52 104 146 10 50 31 40 27 46 Mean=56.2
Difference=30.6

* 1000 random divisions of the 16 numbers into groups of 7 and 9 gave 1000 corresponding values of Difference = Mean A − Mean B. {In other words we permute labels "A" and "B."}

* Of these 126 exceeded Difference = 30.6, for an attained significance level (asl) = .126.



The 14 Scoliosis Patients

Usual Linear Model

y=	$T\beta$ +	X	α +	ϵ
14×1	14×1	14×6	6×1	14×1

where ϵ is distributed as $n(0, \sigma^2 I)$. Usual ANOVA test for H_0: $\beta=0$ rejects H_0 for large values of

$$S = \overset{\perp}{T}' \overset{\perp}{y} / (| \overset{\perp}{T} | | \overset{\perp}{y} |)$$

where $\overset{\perp}{T}$ and $\overset{\perp}{y}$ are projections orthogonal to $\mathcal{L}(X)$ (equivalent to t-test for $\beta = 0$).

Data was actually generated from

$$y = age + .667 \times T + \epsilon$$

where ϵ = -0.16 0.31 2.22 -1.49 -0.66 3.71 2.49 -0.87 -1.37 2.57 -3.47 0.09 -5.23 1.95.

The 14 Scoliosis Patients

			X					
PATIENT	y	T	age	ht	wt	sex	health	constant
[1.]	14.84	1	14.33	175.25	54.35	1	2	1
[2.]	15.65	1	14.67	169.45	53.67	2	0	1
[3.]	19.47	1	16.58	179.35	59.25	1	1	1
[4.]	11.59	-1	13.75	169.85	50.95	1	2	1
[5.]	12.25	-1	13.58	154.75	34.28	1	2	1
[6.]	18.55	1	14.17	173.05	46.45	1	1	1
[7.]	20.33	1	17.17	177.35	54.25	1	2	1
[8.]	12.88	-1	14.42	173.65	59.57	1	0	1
[9.]	10.54	-1	12.58	165.05	39.27	2	0	1
[10.]	17.57	-1	15.67	192.25	64.25	2	1	1
[11.]	16.03	-1	20.17	183.75	62.75	2	2	1
[12.]	18.75	-1	19.33	187.95	56.51	2	1	1
[13.]	9.19	1	13.75	169.05	42.78	1	2	1
[14.]	17.37	1	14.75	177.15	54.35	2	2	1

DATA MATRIX

* Gave S = 0.557. Reject H_0?

* Usual t-test gave asl = P{ $t_7 > 1.77$ } = .060.

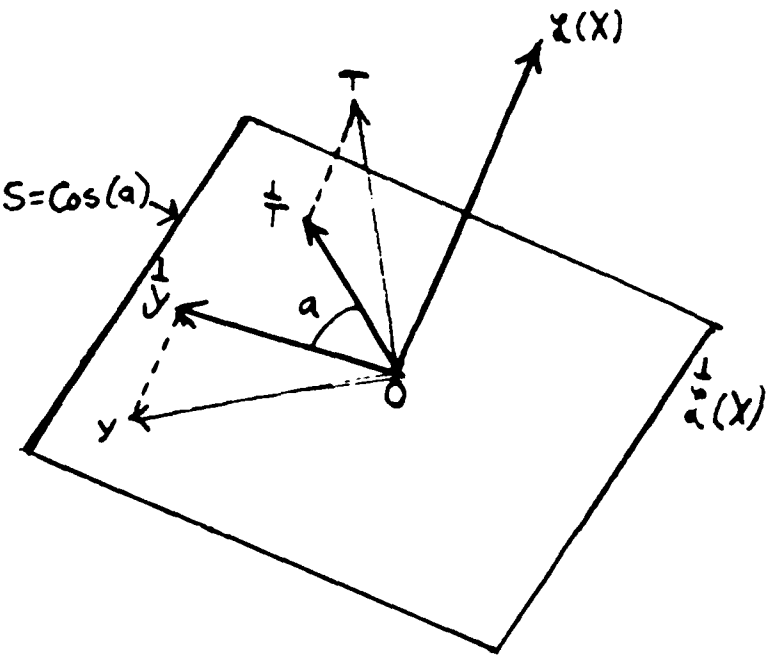
* Compare S with values of S* obtained by permuting T to T* (i.e., permuting -1s and 1s).

* Choose 400 T* vectors randomly from $\binom{14}{7} = 3432$ possibilities.

* Would like $\overset{\perp}{T}{}^*$ to be uniformly distributed in $\overset{\perp}{\mathcal{L}}(X)$.

* For $v_1, v_2, \ldots, v_8$ an orthonormal basis for $\overset{\perp}{\mathcal{L}}(X)$,

looked at projections of $\overset{\perp}{T}{}^*$ vectors along each v_j; counted # projections in deciles of "perfectly uniform" distribution.



counts for 400 Tvecs actually used

decile	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
				↓				
1	33	38	43	31	42	51	45	43
2	37	39	51	66	40	36	33	39
3	31	45	38	57	40	40	43	46
4	38	49	37	29	46	46	36	43
5	39	45	34	12	40	25	46	45
6	45	36	39	11	49	33	36	30
7	47	27	45	28	43	39	45	42
8	42	38	44	67	34	43	37	36
9	34	40	36	67	31	46	34	38
10	54	43	33	32	35	41	45	38
				↑				

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
			↓				
47	40	32	44	41	34	49	33
45	36	39	61	40	49	35	32
43	52	45	57	52	44	37	35
27	36	49	31	41	38	32	44
37	37	37	15	34	38	41	48
44	31	40	11	36	43	32	53
34	42	35	31	38	50	45	32
37	45	51	49	35	35	48	49
44	43	34	56	43	38	38	32
42	38	38	45	40	31	43	42
			↑				

* Projection along v_4 very non-uniform, so choose another 400 T^* vectors.

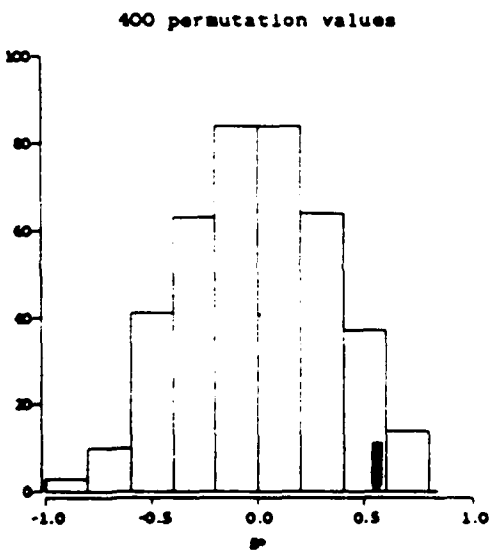
* Not much better, but these are the ones I decided to use.

Each T^* gives $S^* = \overset{\perp}{T}^* \overset{\perp}{y} / (|\overset{\perp}{T}^*| \, |\overset{\perp}{y}|)$ with y fixed as shown. Of the 400 S^* values, 25.5 exceeded S = 0.557 giving asl = 25.5/400 = .064.

* Ideal T vector would have $\tau(T) = |\overset{\perp}{T}|^2 = 14$, that is, $T \perp \mathcal{L}(X)$

* $\mathrm{Var}\{ \hat{\beta} \mid T \} = \sigma^2/\tau(T)$ in usual model.

* If T chosen randomly from 400, mean $\{\tau(T)\} = 8.43$.

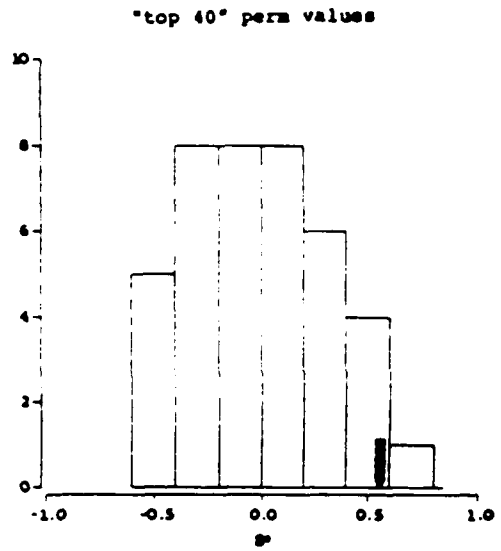


* I chose T randomly from "top 40," i.e. those 40 T vectors having greatest $\tau(T)$ values.
Mean $\{\tau(T)|\text{Top 40}\} = 12.53$.

* Actually got T_{372}, with τ_{372}=12.28.

* 1.5 of the 40 S^* values for the "top 40" reference set exceeded S=0.557, asl = .038.

permutation asl, all 400 = .064 {=25/400}
permutation asl, top 40 = .038 {=1.5/40}
NOTE: Binomial SD for asl is .041
anova asl = .060 {=Prob [$t_7 > 1.77$] }



Maybe top 40 T^* vectors all point in the same direction! No, their direction counts are reasonably uniform in $\overset{\perp}{\mathcal{L}}(X)$, except near v_4. Here the cosines of angle $(\overset{\perp}{y}, v_j)$:

1	2	3	4	5	6	7	8
.31	.19	.01	-.02	.30	.26	-.56	.63

	[.1]	[.2]	[.3]	[.4]	[.5]	[.6]	[.7]	[.8]
[1.]	5	4	4	0	8	2	2	3
[2.]	3	4	4	13	4	8	3	3
[3.]	3	4	3	12	3	4	6	4
[4.]	1	3	5	0	3	3	4	2
[5.]	3	3	4	0	4	4	6	7
[6.]	6	5	2	0	3	3	2	7
[7.]	5	3	6	1	2	6	1	5
[8.]	2	3	3	6	5	4	6	4
[9.]	5	5	4	8	4	5	7	0
[10.]	7	6	5	0	4	1	3	5

The Tau Data

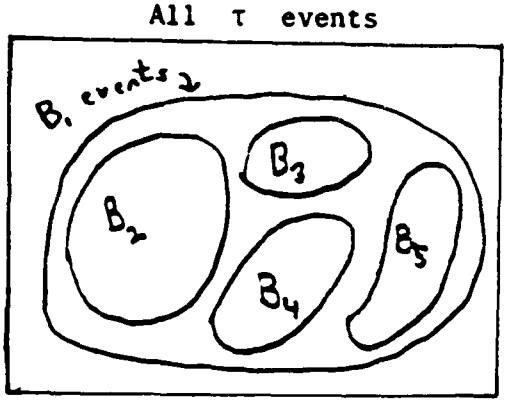
* Occurrence rates of five different tau decay events estimated, B1, B2, B3, B4, B5. (Also the "estimated" SD for each.

* Should have D = B1−(B2+B3+B4+B5) = 0.

* In fact, $\hat{D}$ = 18.25 (or 16.90).

* Wanted: a central 98% confidence interval for D.

* Normal theory: $D \in (15.41, 21.09)$.



* Tried different trimmed means: 0, .1, .2, .25, .3, .5.

* For each trim, evaluated bootstrap SD estimate.

* ".3" gives lowest SD estimate for $\hat{D}$, but ".25" easier to explain.

* Choose ".25" for remainder of analysis.

* Bootstrap SD estimates based on 200 bootstrap replications for each $\hat{B1}$, $\hat{B2}$, ... , $\hat{B5}$.

B1 ESTIMATES (13 LABS)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
EST	84	86.0	85.2	85.2	85.1	87.8	84.7	86.7	86.9	86.1
SD	2	2.2	1.7	2.9	3.1	4.1	1.9	0.7	0.4	1.0

	[,11]	[,12]	[,13]
EST	87.9	87.2	84.7
SD	1.3	0.9	1.0

B2 ESTIMATES (6 LABS)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
EST	24	20.5	22.1	22.3	22.3	22.6
SD	9	4.1	2.5	1.5	2.1	1.5

B3 ESTIMATES (7 LABS)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
EST	9.0	8.0	11.7	9.9	11.8	10.7	10.0
SD	3.8	3.5	1.8	2.1	1.3	0.9	1.8

B4 ESTIMATES (14 LABS)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
EST	18.9	22.4	16.0	18.2	19	17.6	18.3	20.4	13.0	18.2
SD	3.0	5.5	1.3	3.1	9	1.3	3.1	3.3	3.5	0.9

	[,11]	[,12]	[,13]	[,14]
EST	17.4	17.0	18.4	19.1
SD	0.9	1.1	1.6	1.4

B5 ESTIMATES (20 LABS)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
EST	18.3	17.5	22	22.4	15	21	22	18.2	35	17.8
SD	3.0	4.0	10	5.5	3	6	7	3.1	14	2.7

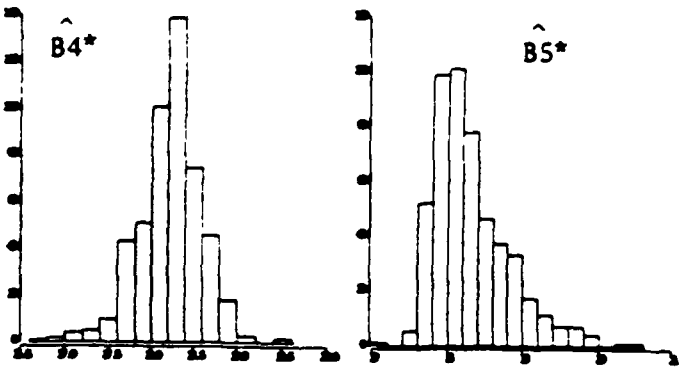
	[,11]	[,12]	[,13]	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]
EST	17.1	17.6	12.9	19.4	18.0	17.7	17.4	18.8	17.7	18.3
SD	1.3	3.3	1.8	2.3	1.2	0.9	1.0	1.1	1.4	1.2

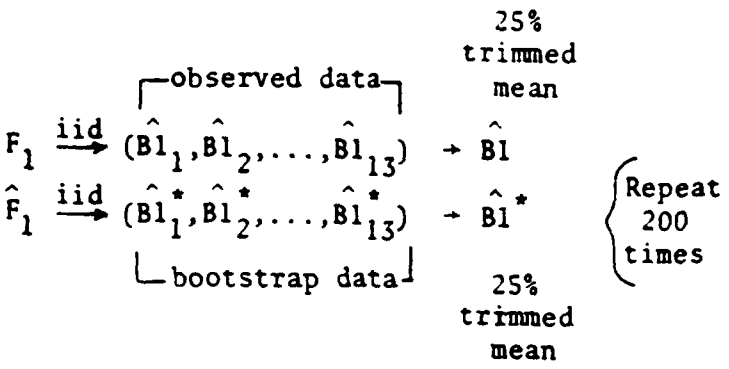
POINT ESTIMATES OF B1,B2,B3,B4,B5 AND D=B1-(B2+B3+B4+B5)

	D	B1	B2	B3	B4	B5
weighted mean:	18.25	86.72	22.31	10.78	17.68	17.70
25%trimmed mean:	16.90	85.88	22.32	10.22	18.26	18.18

matrix of bootstrap sds

TRIM	$\hat{\theta}$	$\hat{B1}$	$\hat{B2}$	$\hat{B3}$	$\hat{B4}$	$\hat{B5}$
0	1.39	0.35	0.43	0.46	0.55	0.94
.1	1.18	0.41	0.41	0.54	0.54	0.69
.2	1.04	0.46	0.40	0.60	0.37	0.54
.25	1.04	0.43	0.39	0.64	0.35	0.54
.3	1.01	0.48	0.38	0.61	0.38	0.49
.5	1.11	0.66	0.37	0.73	0.60	0.36
theo	1.22	.27 small	.86 big	.60	.41	.30 small





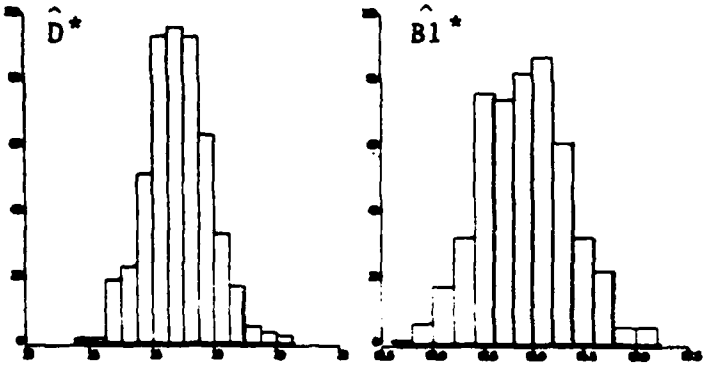
* Then $\hat{SD}(\hat{D}) = [\hat{SD}(\hat{B1})^2 + \cdots + \hat{SD}(\hat{B5})^2]^{\frac{1}{2}}$.

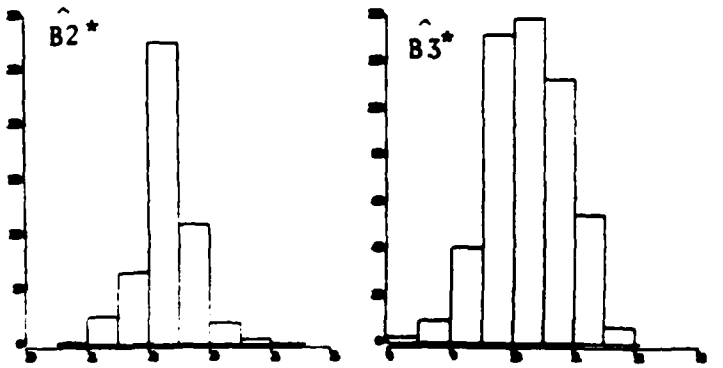
* Here are histograms of $\hat{B1}^*, \ldots, \hat{B5}^*$ and also of $\hat{D}^* = \hat{B1}^* - (\hat{B1}^* + \cdots + \hat{B5}^*)$. 500 boostraps each.

* Percentiles of $\hat{D}^*$ were 14.20 (.01) and 19.34 (.99).

* Approximate confidence intervals for D:

	.01	.99
BC	14.29	19.53
BC_a	14.25	19.49
Boot-T	14.73	18.99
Boot-T (smooth)	14.22	19.20
Normal Theory	15.41	21.09





Bootstrap T

* Let $T = \frac{\hat{D} - D}{\hat{\sigma}}$, where $\hat{\sigma}$ is the jackknife estimate of $SD(\hat{D})$.

* Use bootstrap to estimate $T^{(.01)}, T^{(.99)}$.

* 2000 bootstraps of $T^* = \frac{\hat{D}^* - \hat{D}}{\hat{\sigma}^*}$ gave estimated percentiles

$\hat{T}^{(.01)} = -2.63$ and $\hat{T}^{(.99)} = 2.73$ with

$[\hat{D} - \hat{\sigma} \times 2.73,\ \hat{D} + \hat{\sigma} \times 2.63] = [14.73, 18.99]$

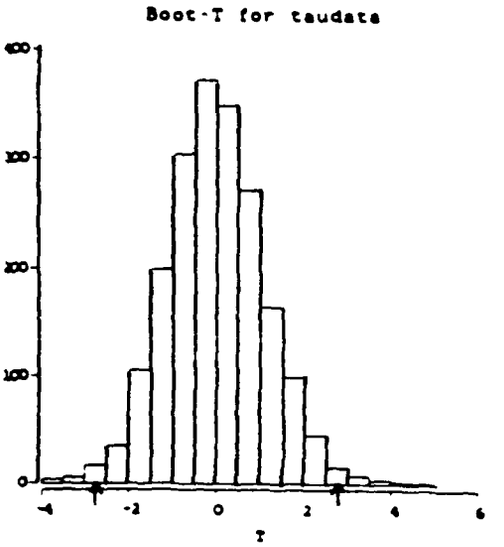
"Boot-T"

* Smoothed bootstrap: Draw from $\tilde{F}_1 = \hat{F}_1 \otimes N(0, \sigma_1^2)$, etc. Gave $\hat{T}^{(.01)} = -2.97$ and $\hat{T}^{(.99)} = 3.31$ so

$[\hat{D} - \hat{\sigma} \times 3.31,\ \hat{D} + \hat{\sigma} \times 2.97] = [14.22, 19.20]$

"Boot-T smooth"

Reference: Efron, B. and Tibshirani, R. (1986). "Bootstrap methods for standard errors, confidence intervals and other measures of statistical accuracy," *Statistical Science*, 1, 54-77.



Cancer Treatment Data

NCOG7b61 SEPARATED AND ORDERED

	group A					group B				
	y	d	km	ent	ind	y	d	km	ent	ind
1	7	1	0.980	1131.188	24	37	1	0.978	160.190	54
2	34	1	0.961	1795.380	44	84	1	0.956	960.566	73
3	42	1	0.941	2096.754	48	92	1	0.933	1745.942	88
4	63	1	0.922	861.252	16	94	1	0.911	979.004	72
5	64	1	0.902	1210.064	27	110	1	0.889	1588.752	85
6	74	0	0.902	1117.750	25	112	1	0.867	605.316	61
7	83	1	0.882	1529.876	38	119	1	0.844	2316.252	96
8	84	1	0.862	849.814	15	127	1	0.822	831.814	71
9	91	1	0.842	472.564	6	130	1	0.800	439.126	58
10	108	1	0.822	972.566	12	133	1	0.778	1684.066	87
11	112	1	0.802	1818.818	43	140	1	0.756	720.068	66
12	129	1	0.782	1812.818	42	146	1	0.733	1333.816	78
13	133	1	0.762	1683.066	39	155	1	0.711	1264.940	77
14	133	1	0.742	880.252	17	159	1	0.689	1034.442	76
15	139	1	0.722	1431.568	34	169	0	0.689	472.564	60
16	140	1	0.702	690.630	10	173	1	0.666	1370.692	79
17	140	1	0.681	968.566	20	179	1	0.643	291.942	52
18	146	1	0.661	714.068	11	194	1	0.620	810.376	70
19	149	1	0.641	1390.692	35	195	1	0.597	2314.252	95
20	154	1	0.621	2068.316	45	209	1	0.574	1991.440	91
21	157	1	0.601	671.630	9	249	1	0.551	1567.314	84
22	160	1	0.581	349.818	2	281	1	0.528	618.754	63
23	160	1	0.561	797.376	13	319	1	0.505	720.068	65
24	165	1	0.541	1719.504	41	339	1	0.482	1677.066	86
25	173	1	0.521	2106.192	46	432	1	0.459	1006.442	74
26	176	1	0.501	1694.066	40	469	1	0.436	1381.692	81
27	185	0	0.501	1392.692	32	519	1	0.413	2105.192	92
28	218	1	0.480	892.690	18	528	0	0.413	438.126	57
29	225	1	0.459	2150.630	50	547	0	0.413	2153.630	94
30	241	1	0.438	528.440	7	613	0	0.413	2106.192	93
31	248	1	0.418	171.190	1	633	1	0.386	748.500	67
32	273	1	0.397	802.376	14	725	1	0.358	1857.688	89
33	277	1	0.376	321.380	4	759	0	0.358	1968.002	90
34	279	0	0.376	2380.128	51	817	1	0.328	410.688	59
35	297	1	0.354	1409.130	33	1092	0	0.328	1526.876	83
36	319	0	0.354	202.628	3	1245	0	0.328	1357.254	80
37	405	1	0.330	887.690	19	1331	0	0.328	1398.692	82
38	417	1	0.307	2201.500	49	1557	1	0.287	605.316	62
39	420	1	0.283	1517.438	37	1642	0	0.287	1084.318	75
40	440	1	0.259	349.818	5	1771	0	0.287	818.376	68
41	523	0	0.259	2153.630	47	1776	1	0.230	407.688	53
42	523	1	0.233	1041.880	21	1897	0	0.230	774.938	69
43	583	1	0.208	1480.000	36	2023	0	0.230	621.754	64
44	594	1	0.182	546.440	8	2146	0	0.230	338.818	55
45	1101	1	0.156	1037.880	23	2297	0	0.0	431.126	56
46	1116	0	0.156	1259.940	28					
47	1146	1	0.125	1320.816	30					
48	1226	0	0.125	1040.880	22					
49	1349	0	0.125	1293.378	31					
50	1412	0	0.125	1314.816	29					
51	1417	1	0.0	1172.626	26					

y=days observed
d=failure or not
km=Kaplan-Meier
ent=entry date (from 1/1/78)
ind=index number in original data

* Randomized clinical trial for head and neck cancer.

* Data as of June 30, 1985.

* 51 patients in "A," radiation.

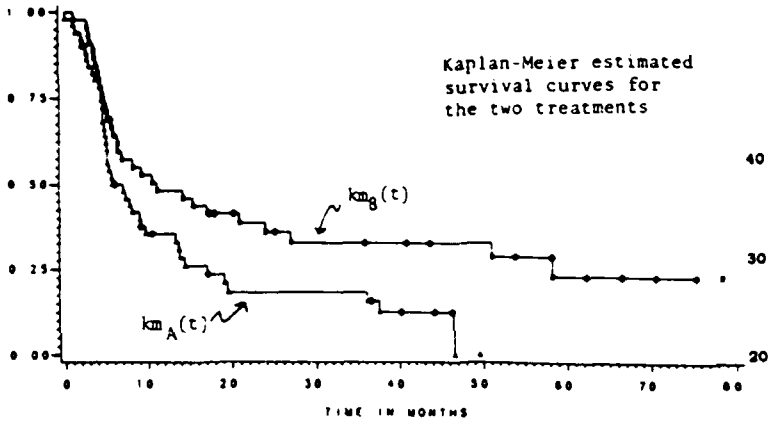
* 46 patients in "B," radiation plus chemotherapy.

* y = time to relapse (days).

* $d = \begin{cases} 1 & \text{if relapse} \\ 0 & \text{censored} \end{cases}$ observed.

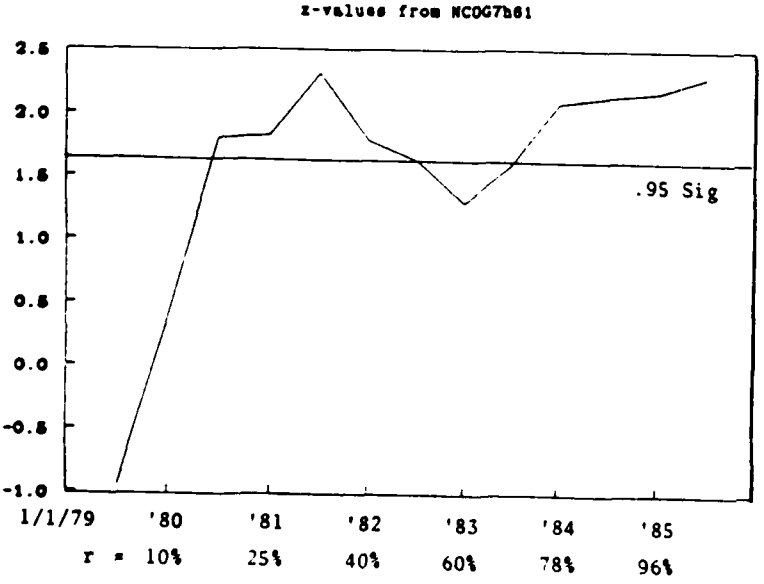
* km = Kaplan-Meier survival curve.

* Log-rank (Mantel-Haenszel) test for equality of survival was z = 2.29 for an attained level of significance $1\text{-}\Phi(2.29) = .011$.



* z-value of log-rank test at various calendar times.

* z = 2.31 on 6/30/81. Experiment nearly halted.



* "r" = proportion of total experience ($\sum$ y for all patients at that date compared to $\sum$ y on June 30, 1985).

* Question: Was treatment B relatively more effective early in the experiment?

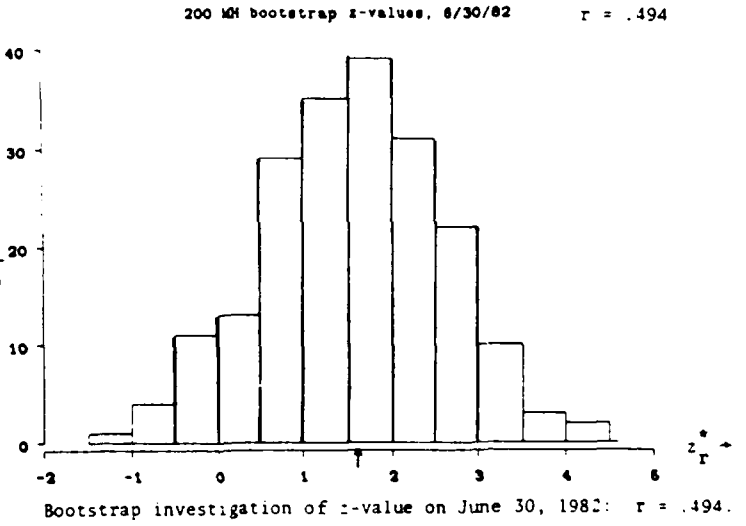
Some dubious theory: Let "z_r" be the z-value when the proportion r of the total data is available (so z_1 = final z-value). Then

(a) $E\{z_r\}/E\{z_1\} \doteq \sqrt{r}$

(b) $z_r \dot{\sim} N(\sqrt{r}\, E\{z_1\}, 1)$

(c) $z_r \mid z_1 \dot{\sim} N(\sqrt{r}\, E\{z_1\}, 1 - r)$

(d) z_{r_1} and z_{r_2} are approximately bivariate normal with corr $= \sqrt{\frac{r_1}{r_2}}$.



* Consider as fixed the 72 entry dates (38 for A, 34 for B) occurring before June 30, 1982.

* For entry date e_i compute c_i = # of days from e_i until June 30, 1982.

* Let $Y_1^*, Y_2^*, \ldots, Y_{38}^*$ be i.i.d. draws from km_A, the *final* Kaplan-Meier curve as of June 30, 1985.

* For each Y_i^*, let $y_i^* = \min(Y_i^*, c_i)$ and $d_i^* = 1$ or 0 as $y_i^* = Y_i^*$ or c_i.

* Likewise draw $Y_1^*, Y_2^*, \ldots, Y_{34}^*$ from km_B, the final Kaplan-Meier curve as of June 30, 1985.

* Then compute z_r^*, the log-rank z-value for the bootstrap data.

* 200 $z_r^{*\prime}$s $\sim$ N(1.56, 1.04^2), $\sqrt{r}\ E\{z_1\} = 1.61$

* Compare with N(1.61, 1) from (b) !

* Did correspomding $z_r^{*\prime}$s for r = .246 (January 1, 1981)?

* Corr (z_{r_1}, z_{r_2}) = .727.

* Compare with (d), corr = $\sqrt{\frac{r_1}{r_2}} = .706$.

* Jagged line: z_r versus r.

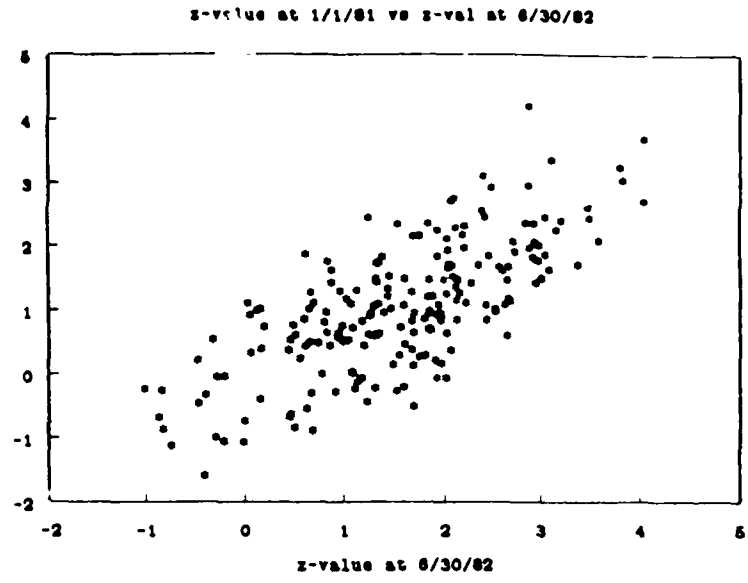
* Smooth curves:

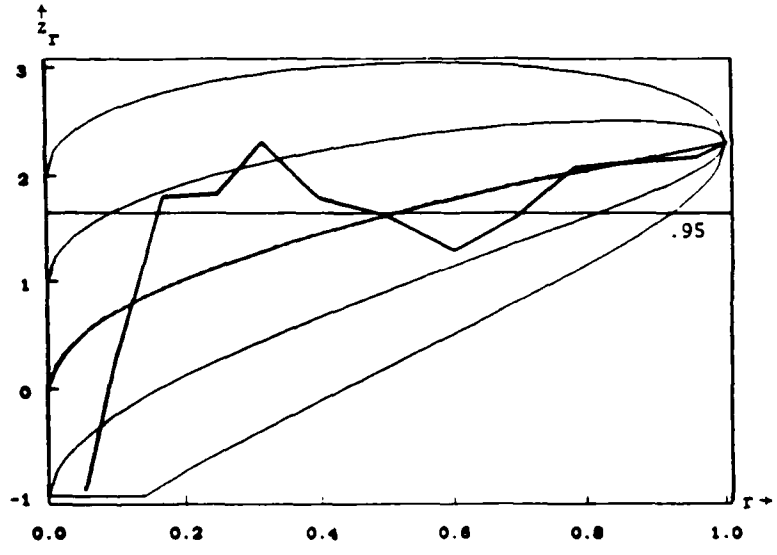
$\sqrt{r}\ z_1 + c\ \sqrt{1-r}$, c = .2, −1, 0, 1, 2.

* Middle curve is E { $z_r \mid z_1$}.

* Others show 1 and 2 conditional deviation excursions.

* Nothing unusual happened! Maximum excursion is less than 1.3 conditional standard deviations.





II. SPECIAL INVITED PAPER

Fitting Functions to Noisy Data in High Dimensions

Jerome H. Friedman, Stanford University

FITTING FUNCTIONS TO NOISY DATA IN HIGH DIMENSIONS

Jerome H. Friedman*
Department of Statistics
Stanford Linear Accelerator Center
Stanford University

Abstract

Consider an arbitrary domain of interest in n-dimensional Euclidean space and an unknown function of n arguments defined on that domain. Suppose we are given the value of the function (perhaps perturbed with additive noise) at some set of points. The problem is to find a function that provides a reasonable approximation to the unknown one over the domain of interest. This paper presents a brief review of current methodology aimed at dealing with this problem, and presents a new technique – multivariate adaptive regression splines – that has the potential to overcome some of the limitations of previous approaches.

1.0. Introduction

Suppose a system under study can be described (over some domain $D \in R^n$) by

$$y = f(x_1, \cdots, x_n) + \epsilon \tag{1}$$

where y is a response or dependent variable of interest, $x_1, \cdots, x_n$ are a set of explanatory or independent variables, and f is a (deterministic) single valued function of its n-dimensional argument. The quantity ϵ is an additive random or stochastic component that (if nonzero) reflects the fact that y depends on quantities other than $x_1 \cdots x_n$ that are also varying. We are given a set of values $\{y_i, x_{1i}, \cdots, x_{ni}\}_1^N$, $(x_{1i}, \cdots, x_{ni}) \in D$, (training sample) and the purpose of the exercise is to obtain a function $\hat{f}(x_1, \cdots, x_n)$ that provides a reasonable approximation to $f(x_1, \cdots, x_n)$. Here reasonable usually means accurate since one often wants to use $\hat{f}$ to approximate f at other points not part of the training sample. If in addition one wants to use $\hat{f}$ to try to understand the properties of f (and thereby the system that provided the data) then the interpretability of the representation of $\hat{f}$ is important. It is also sometimes important that $\hat{f}$ be rapidly computable. In addition, for some applications it is important that $\hat{f}$ be a smooth function of its argument; that is, at least its low order derivatives exist everywhere in D.

* Research supported in part by National Security Agency Grant MDA904-88-H-2029

In low dimensional settings ($n \leq 2$) successful developments have occurred in two general directions: piecewise polynomials and local averaging. The basic idea of piecewise polynomials is to approximate f by several generally low order polynomials each defined over a different subregion of the domain D. The approximation is required to be continuous, and sometimes have continuous low order derivatives. The tradeoff between smoothness and flexibility of the approximation $\hat{f}$ is controlled by the number of subregions (knots) and the order of the lowest derivative allowed to be discontinuous at region boundaries. The most popular piecewise polynomial fitting procedures are based on splines. [See deBoor (1978) for a general review of splines and Schumacker (1976), (1984) for reviews of some two-dimensional extensions.]

Local averaging approximations take the form

$$\hat{f}(x) = \sum_{i=1}^{N} K(x, x_i)\, y_i \tag{2}$$

where $K(x, x')$ (called the kernel function) usually has its maximum value at $x' = x$ with its absolute value decreasing as $|x - x'|$ increases. Thus, $\hat{f}(x)$ is taken to be a weighted average of the y_i where the weights are larger for those observations that are close or local to x. For $n > 1$ the kernel is usually taken to be a function of the Euclidean distance between the points

$$K(\mathbf{x}, \mathbf{x}') = K\left[\left(\sum_{i=1}^{n} |x_i - x_i'|^2\right)^{1/2}\right] \tag{3}$$

Local averaging procedures have received considerable attention in the statistical literature beginning with their introduction by Parzen (1962). Stone (1977) has shown that this approach has desirable asymptotic properties. They have also seen interest from the mathematical approximation literature [Shepard (1964), Bozzini and Lenarduzzi (1985)]. Roughness penalty methods [smoothing ($n = 1$) and thin plate ($n = 2$) splines] are closely related to kernel methods based on Euclidean distance [see Silverman (1985) and Schumaker (1976)].

The direct extension of piecewise polynomials (splines) or local averaging methods to higher dimensions ($n > 2$) is straightforward in principle but difficult in practice. These difficulties are related to the so-called "curse-of-dimensionality", a phrase coined by Bellman (1961) to express the fact that exponentially increasing numbers of points are needed to densely populate Euclidean spaces of increasing dimension. In the case of spline approximations, extension to higher dimensions is accomplished through tensor products of univariate spline functions. These functions are associated with a grid of points defined by the outer product of knot positions on each independent variable. For a given number of knots K on each variable, the size of the grid, and thus the number of approximating basis functions, grows as K^n. For example, in six dimensions a (tensor product) cubic spline with only *one* interior knot in each variable has 15,625 coefficients to be estimated. That number in ten dimensions is approximately 10^7. Even though only one interior knot per variable might be considered a very coarse grid, it still requires a very large number of data points to estimate the corresponding spline approximation. Finer grids require many more points.

Local averaging methods suffer a similar fate as the dimension of the function argument space increases. For example, let D be the unit hypercube in R^n and consider a uniform kernel with hypercubical support and bandwidth (edge length) covering 10 percent of the range of each coordinate. Then, if the data are roughly uniformly distributed in R^n, the kernel will (on average) contain only $(0.1)^n$ of the sample, thereby nearly always being empty for moderate to large n. If, on the other hand, one adjusts the size of the neighborhood (bandwidth) to contain 10 percent of

the sample, it will cover (on average) $(0.1)^{1/n} \times 100$ percent of the range of each variable, resulting in a very crude approximation.

This problem of the inherent sparsity of practical sampling in high dimensions basically limits the straightforward application of both piecewise polynomials and local averaging methods in these settings. It does not, however, limit theoretical investigation. It is straightforward to imagine arbitrarily densely sampling of high dimensional spaces. Asymptotic theoretical calculations can then be done. [See Stone (1977) for pioneering work in this area.] The (practical) difficulty lies only in obtaining the corresponding large samples required for accurate approximations. It should be noted in addition, that local averaging approximations (and to a lesser extent tensor product splines) are slow to compute and difficult to interpret.

The curse-of-dimensionality is fundamental and cannot be directly overcome. If the true underlying function $f(x_1, \cdots, x_n)$ (1) exhibits strong variation of no special structure on all of the variables in every part of the domain D, then accurate approximation with feasible sample sizes is not possible. Fortunately, very few functions of interest exhibit behavior quite this dramatic. Generally there is some (sometimes known, more often unknown) special structure associated with the function that can be exploited by a sufficiently clever algorithm to reduce the complexity and thereby achieve more accurate approximation.

Function approximation in high dimensional settings has been pursued mainly in statistics. The principal approach taken there has been to fit an especially simple parametric form to the training sample. The most common parameterization is the linear function

$$\hat{f}(x_1, \cdots, x_n) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i. \tag{4}$$

This is not likely to produce a very accurate approximation to very many functions in R^n, but it has the virtue of requiring relatively few data points, it is easy to interpret, and it is rapidly computable. Also, if the stochastic component ϵ (1) is large compared to f, then the variability of the estimate dominates, and the systematic error associated with this simple approximation is not the most serious problem.

Recently, the linear model has been generalized nonparametrically to the so-called additive model

$$\hat{f}(x_1, \cdots, x_n) = \sum_{i=1}^{n} f_i(x_i) \tag{5}$$

[Friedman and Stuetzle (1981), Breiman and Friedman (1985), Hastie and Tibshirani (1986), Friedman and Silverman (1987)]. Here the $\{f_i(x_i)\}_1^n$ are each (different) smooth but otherwise arbitrary functions of a single variable. Although additive models are still not able to accurately approximate very general functions in R^n, they do constitute a much richer class than the simple linear approximation (4). They share the high interpretability of the linear model (one can view the univariate functions f_i) and they are not overly difficult to compute.

Linear and additive approximations lack generality in that they have limited ability to adapt to a wide variety of multivariate functions f. Also, as the sample size increases there is a limit to the accuracy of the approximation (unless the true underlying function happens to be exactly linear or additive over D).

Strategies that attempt to approximate general functions in high dimensionality are based on adaptive computation. An adaptive computation is one that dynamically adjusts its strategy to take into account the behavior of the particular problem to be solved, e.g. the behavior of the function to be approximated. Adaptive algorithms have been in long use in numerical quadrature [see Lyness (1970); Friedman and Wright (1987).] In statistics, adaptive algorithms for function

approximation have been developed based on two paradigms, recursive partitioning [Morgan and Sonquist (1963), Breiman, Friedman, Olshen, and Stone (1984)], and projection pursuit [Friedman and Stuetzle (1981), Friedman, Grosse, and Stuetzle (1983), Friedman, (1985)].

Projection pursuit uses an approximation of the form

$$\hat{f}(x_1, \cdots, x_n) = \sum_{m=1}^{M} f_m \left(\sum_{i=1}^{n} \alpha_{im} x_i \right), \tag{6}$$

that is, additive functions of linear combinations of the variables. The univariate functions, f_m, are required to be smooth but are otherwise arbitrary. These functions, and the corresponding coefficients of the linear combinations appearing in their arguments, are jointly optimized to produce a good fit to the data based on some distance (between functions) criterion - usually squared-error loss. It can be shown [see Diaconis and Shahshahani (1984)] that any smooth function of n variables can be represented by (6) for large enough M. The effectiveness of the approach lies in the fact that even for small to moderate M, many classes of functions can be closely fit by approximations of this form[see Donoho and Johnstone (1985).] Another advantage of projection pursuit approximations is affine equivariance. That is, the solution is invariant under any nonsingular affine transformation (rotation and scaling) of the original explanatory variables. It is the only general method suggested for practical use that seems to possess this property. Projection pursuit solutions have some interprative value (for small M) in that one can inspect the functions f_m and the corresponding linear combination vectors. Evaluation of the resulting approximation is computationally fast. Disadvantages of the projection pursuit approach are that there exist some simple functions that require large M for good approximation [see Huber (1985)], it is difficult to separate the additive from the interaction effects associated with the variable dependencies, interpretation is difficult for large M, and the approximation is computationally time consuming to construct.

Recursive partitioning approximations take the form

$$\hat{f}(x_1, \cdots, x_n) = \sum_{m=1}^{M} f_m(x_1, \cdots, x_n) I[(x_1, \cdots, x_n) \in R_m]. \tag{7}$$

Here $I(\cdot)$ is 0/1 valued function that indicates the truth of its argument and $\{R_m\}_1^M$ are disjoint subregions representing a partition of D. The functions f_m are generally taken to be of quite simple parametric form. The most common is a constant function

$$f_m(x_1, \cdots, x_n) = a_m \tag{8}$$

[Morgan and Sunquist (1963) and Breiman, et al. (1984)]. Linear functions (4) have also been proposed [Breiman and Meisel (1976) and Friedman (1979)], but they have not seen much use. The partitioning is developed in a recursive manner. At each step, M, all existing subregions $\{R_m\}_1^M$ are optimally split into two subregions along one of the variables. The particular split that yields the best improvement in the fit is taken to define two new regions and the parent region (that was split) is deleted. (The starting region is the entire domain D.) The number of subregions in the partition is thereby increased by one at each step. A backwards stepwise strategy for determining the final number of regions is detailed in Breiman, et al. (1984).

The recursive partitioning approach has the potential to provide acceptable approximations in high dimensionalities provided the under'ying function has low "local" dimensionality. That is, even though the function f (1) may strongly depend on all of the variables, in any local region of the domain the dependence is strong on only a few of them. These few variables may be different

in different regions. Another assumption inherent in the recursive partitioning strategy is that interaction effects have marginal consequences. That is, a local intrinsic dependence on several variables, when best approximated by an additive function, does not lead to a constant model. This is nearly always the case.

Recursive partitioning using piecewise constant approximations (8) are fairly interpretable owing to the fact that they are very simple and can be represented by a binary tree. [See Breiman et al. (1984)]. They are also fairly rapid to construct and especially rapid to evaluate.

Although recursive partitioning is the most adaptive of the methods for multivariate function approximation it suffers from some fairly severe restrictions that limit its effectiveness. Foremost among these is that the approximating function is discontinuous at the subregion boundaries. This is more than a cosmetic problem. It severely limits the accuracy of the approximation, especially when the true underlying function is continuous. Even imposing continuity only of the function (as opposed to derivatives of low order) is usually enough to dramatically increase approximation accuracy.

Another problem with recursive partitioning is that certain types of simple functions are difficult to approximate. These include linear functions with more than a few nonzero coefficients [with the piecewise constant approximation (8)] and additive functions (5) in more than a few variables (piecewise constant or piecewise linear approximation). In addition, one cannot discern from the representation of the model whether the approximating function is close to a simple one, such as linear or additive, or whether it involves complex interactions among the variables.

2.0. Multivariate Adaptive Regression Splines.

This section describes a new method of adaptive computation for approximating functions in high dimensionalities. Although it is an extension of the additive modeling (5) procedure developed by Friedman and Silverman (1987), it appears closest in spirit to the adaptive nature of the recursive partitioning approach. Unlike recursive partitioning, however, it produces strictly continuous approximations (with continuous derivatives if desired), it easily approximates linear and additive functions, and it can be represented in a form that permits separate identification of the additive and (multiple) interaction effects associated with the variables that enter into the model.

The approximation takes the form of an expansion in multivariate spline basis functions,

$$\hat{f}(x_1, \cdots, x_n) = \sum_{m=0}^{M} a_m B_m(x_1, \cdots, x_n) \tag{9a}$$

with

$$B_0(x_1, \cdots, x_n) = 1, \tag{9b}$$

$$B_m(x_1, \cdots, x_n) = \prod_{k=1}^{K_m} b(x_{v(k,m)}|t_{km}), \quad m \geq 1. \tag{9c}$$

The $\{a_m\}_0^M$ are the coefficients of the expansion. Each multivariate spline basis function B_m, $m > 0$, is a product of univariate spline basis functions b, each of a single variable $x_{v(k,m)}$, characterized by a knot at t_{km}. The subscripts $v(k,m)$ label the explanatory variables, thereby taking values in the range $1 \leq v(k,m) \leq n$; K_m takes values in the same range $1 \leq K_m \leq n$ and determines the number of factors (univariate spline basis functions) comprising the corresponding B_m. The multivariate spline basis functions B_m are adaptive in that the number of factors K_m, the variable set $V(m) = \{v(k,m)\}_1^{K_m}$ and the knot set $\{t_{km}\}_1^{K_m}$ are all determined by the data.

The approximation is developed in a forward/backwards stepwise recursive manner in analogy with the recursive partitioning approach. Given $\{B_m\}_0^{M-1}$ the Mth term takes the form

$$B_M(x_1, \cdots, x_n) = B_\ell(x_1, \cdots, x_n) b(x_v|t) \tag{10}$$

with $0 \leq \ell \leq M-1$. That is, the next term B_M is taken to be the product of a univariate spline basis function with one of the previously defined multivariate spline basis functions B_ℓ $(0 \leq \ell \leq M-1)$. The values for v, t, and ℓ are chosen so as to jointly maximize the goodness-of-fit of the resulting approximation (see Section 2.2). The defining variable x_v for the new basis function $b(x_v|t)$ is restricted to be one that does not appear in the selected B_ℓ, so that the same variable does not appear more than once in any B_m $(0 \leq m \leq M)$. The resulting optimal values v^*, t^*, and ℓ^* are then used to form the new multivariate spline basis function

$$B_M = \prod_{k=1}^{K_M} b(x_{v(k,M)}|t_{kM})$$

with $K_M = K_{\ell^*} + 1$, $v(K_M, M) = v^*$, $t_{K_M M} = t^*$, and the rest of the factors taken from B_{ℓ^*}.

One of the requirements for this strategy to be computationally feasible is that each univariate basis function be defined by the location of a single knot t_{km}. We therefore use the truncated power basis representation for the (univariate) splines

$$b^{(q)}(x|t) = (x-t)_+^q \tag{11}$$

where q is the order of the spline which controls the degree-of-continuity of the approximation. The subscript denotes the non-negative part. (This basis is known to produce numerical problems, especially for $q > 1$, so a great deal of care must be taken in the implementation.)

This forward stepwise construction of the multivariate spline basis (9) (10) is continued until $M = M_{\max}$ terms have been entered into the approximation. This process yields a sequence of $M_{\max}$ models, each with one more term than the previous one in the sequence. Each model in the sequence has an associated badness-of-fit score (see Section 2.2). That model with the lowest badness-of-fit score is then subjected to a backwards stepwise deletion strategy [see Friedman and Silverman (1987), Section 2.1], to obtain the final model. The upper limit $M_{\max}$ should be taken to be large enough so that the minimizing model is not too close to the end of the sequence. Due to the forward stepwise nature of the procedure it is possible for the badness-of-fit to locally increase a bit as the sequence proceeds, and then start to decrease again.

If one makes the restriction $K_m = 1$ (9c) for all m (that is, always setting $\ell = 0$ rather than including it in the optimization) the approximation becomes a sum of functions, each of a single variable. This is, of course, an additive model (5) and this strategy reduces to the smoothing and additive modeling technique introduced by Friedman and Silverman (1987). The key ingredient that advances this approach to general settings is the ability to fit (possibly complex) interactions among the variables through the product terms that are permitted to enter the approximation (9), if required by the fit.

Although originally motivated by the work of Friedman and Silverman (1987) this approximation strategy (9)-(11) has more in common with the recursive partitioning approach (see Section 1.0) to function approximation (7). There is a correspondence between the terms in (9) and the regions in (7). Choosing a previous term for multiplication (10) is analogous to choosing a (previous) region to split in (7). The optimization over v and t in (10) is quite similar to finding the optimal splitting variable and split point for partitioning a region.

The correspondence between this basic approach and recursive partitioning is most easily seen by contrasting the piecewise constant approximation (8) of the latter with the use of $q = 0$ splines (11) in the former

$$b^{(0)}(x|t) = I(x-t). \tag{12}$$

Both methods then produce piecewise-constant approximations in this case, and multiplying (sometimes with constraints) is strictly equivalent to splitting. The two methods, even though being most

similar in this setting, do not however produce equivalent approximations. This is basically because unlike recursive partitioning, the subregions induced by (9), (10), (12) are not constrained to be disjoint. At any stage during recursive partitioning, only terminal regions are eligible for splitting, i.e. only those regions defined by the intersections of previous splits (terminal nodes on the current binary tree). With the MARS strategy all previously defined regions – not just terminal ones – are eligible for splitting at any stage of the model building process. The previously defined regions are those represented by the internal nodes of the tree and are unions of subsets of current terminal regions.

The strategy associated with the MARS approach has several important advantages. Foremost among them is that it allows close approximations to many of the common functions that present difficulty to recursive partitioning (e.g. nearly linear or additive functions). Another advantage is its interpretability through its ANOVA representation (see below). The most important advantage of this approach, however, is that by choosing $q > 0$ (11) continuous approximations can be achieved. This has been one of the most serious limitations of recursive partitioning. Choosing a value for $q \geq 1$ causes the approximation to be continuous and to possess continuous derivatives to order $q - 1$.

As with recursive partitioning, this method attempts to use to advantage the fact that interaction effects involving several variables will give rise to non-constant dependencies on at least one of those variables individually. This is because in the forward part of the model building strategy, additive terms and lower order interactions must enter before the corresponding higher order interactions. These lower order terms provide information as to where to place knots to capture the corresponding higher order ones, and they may in fact be removed (through the backwards deletion process) after the higher order interaction terms are entered.

2.1. ANOVA Decomposition.

The representation of the approximation given by (9), (10), (11) resulting from construction of the model

$$\hat{f}(x_1, \cdots, x_n) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K_m} b(x_{v(k,m)} - t_{km})_+^q \tag{13}$$

does not provide much insight into the nature of the approximation. By simply rearranging the terms, however, it is able to provide considerable insight into the predictive relationship between y and $x_1, \cdots, x_n$,

$$\begin{aligned} \hat{f}(x_1, \cdots, x_n) = a_0 &+ \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) \\ &+ \sum_{K_m=3} f_{ijk}(x_i, x_j, x_k) + \cdots. \end{aligned} \tag{14a}$$

Here the first sum is over all terms involving only a single variable and represents the purely additive component of the model. Each additive function $f_i(x_i)$ can be computed by collecting together all single variable terms involving x_i,

$$f_i(x_i) = \sum_{\substack{K_m=1 \\ i \in V(m)}} a_m B_m(x_i). \tag{14b}$$

Here $V(m)$ represents the variable set $\{v(k,m)\}_1^{K_m}$ associated with the mth term. The second sum in (14a) is over all terms involving exactly two variables and represents the pure first order (two variable) interaction part of the model with

$$f_{ij}(x_i, x_j) = \sum_{\substack{K_m=2 \\ (i,j) \in V(m)}} a_m B_m(x_i, x_j). \tag{14c}$$

Similarly, the third sum represents second order (three variable) interactions with

$$f_{ijk}(x_i, x_j, x_k) = \sum_{\substack{K_m=3 \\ (i,j,k)\in V(m)}} a_m B_m(x_i, x_j, x_k), \tag{14d}$$

and so on. The additive terms can be viewed by plotting $f_i(x_i)$ against x_i as one does with additive modeling. The two variable interaction terms $f_{ij}(x_i, x_j)$ can be plotted using either contour or perspective mesh plots. Higher order interactions (if present) are of course more difficult to view. The corresponding (multivariate) knot locations can, however, provide some insight. We refer to (14) as the ANOVA decomposition or representation of the MARS model because of its similarity to decompositions provided by the analysis of variance of contingency tables.

The ANOVA representation identifies the particular variables that enter into the model, whether they enter purely additively or are involved in interactions with other variables, the order of the interactions, and the other variables that participate in them.

2.2. Model Selection.

As in Friedman and Silverman (1987) we use the generalized cross-validation criterion (Craven and Wahba, 1979)

$$GCV(M) = \frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}_M(x_{1i}, \cdots, x_{ni})]^2 / \left[1 - \frac{C(M)}{N}\right]^2 \tag{15a}$$

for model selection where M is the number of terms in (9a) and

$$C(M) = (d+1)M + 1. \tag{15b}$$

Minimization of this criterion is used to select the knot variable and its location at each forward step, the terms to delete in the backwards steps, and the size of the final model. The use of (15b) results in a change of $(d+1)$ "degrees-of-freedom" for each term in the model, one for fitting the least-squares coefficient a_m, and d for the optimization associated with the knot placement. Friedman and Silverman (1987) used $d = 2$. This was motivated somewhat on theoretical grounds but mostly on an empirical basis. This value is too small for generalized MARS modeling since we are, in addition, optimizing over the term index $0 \le \ell \le M-1$ at each step as well as the knot location. This produces increased variance that must be accounted for in the model selection. A direct approach would be to estimate an optimal d value for the problem at hand through a sample reuse technique such as the 632 bootstrap (Efron, 1983) or cross-validation Stone (1974).

Another approach is to study the variance directly through a modified bootstrapping technique (Hastie and Tibshirani, 1985). Each bootstrap replication consists of replacing each response value by a standard normal deviate. By construction the true underlying function f is the constant zero, and the mean-squared-prediction error is completely dominated by the variance

$$E(f - \hat{f}_M)^2 = E\hat{f}_M^2 = \text{Var}\ \hat{f}_M$$

or equivalently

$$E(y - \hat{f}_M)^2 = E\hat{f}_M^2 + 1. \tag{16}$$

Since the GCV score (15a) is intended to be an estimate for (16) one can obtain an estimate for $C(M)$ through

$$E(ASR_M)/\left[1 - \frac{\hat{C}(M)}{N}\right]^2 = E\hat{f}_M^2 + 1$$

or

$$\hat{C}(M) = N\left[1 - \left(\frac{E(ASR_M)}{Ef_M^2 + 1}\right)^{1/2}\right] \tag{17}.$$

Here the average-squared-residual, ASR_M, is the numerator in (15a). The expected values in (17) are estimated through repeated bootstrap replications.

A wide variety of simulation studies (not detailed here) using this approach indicate the following.

(1) $C(M)$ is a monotonically increasing function with decreasing slope as M increases.
(2) Using the linear approximation (15b), with $d = 2.5$, is fairly effective, if somewhat crude.
(3) The "best" value for d depends (weakly) on M, N, and the distribution of the covariate vectors.
(4) Over a wide variety of situations, the best value of d lies in the range $2.0 \leq d \leq 3.0$.
(5) The *actual* accuracy of the approximation, in terms of integrated squared error

$$ISE = \int [f(\mathbf{x}) - \hat{f}(\mathbf{x})]^2 dF(\mathbf{x}),$$

depends very little on the value chosen for d in the range $2.0 \leq d \leq 3.0$.
(6) The *estimated* accuracy

$$E[ISE - GCV(M^*)]^2,$$

with M^* being the minimizer of (15), does show a moderate dependence on the choice of d.

The consequence of (5) and (6) is that, although how well one is doing with this approach is fairly independent of d, how well one *thinks* he is doing (based on the optimizing GCV score) does depend somewhat on the values chosen for d. Therefore, a sample reuse technique should be used to estimate the predictive capability of the final model, if it needs to be known fairly precisely.

2.3. Degree-of-Continuity.

Another important choice is the degree of continuity to be imposed on the approximating function, i.e. the value for q in (11). This choice affects the accuracy of the approximation, and the speed and numerical stability of the computation. Friedman and Silverman (1987) used $q = 1$ in conjunction with the knot placement and model selection strategy. This produces a continuous piecewise linear approximation with discontinuous derivatives. Advantages of this approach are much more rapid and numerically stable computation compared to higher values of q. Also, it can provide more accurate approximations in some situations. The main disadvantage is discontinuous first derivatives.

Friedman and Silverman (1987) provide for derivative smoothing by replacing the basis functions $b^{(1)}(x|t)$ (11) by closely related ones with continuous first derivatives:

$$C(x|t_-, t, t_+) = \begin{cases} 0 & x \leq t_- \\ p(x - t_-)^2 + r(x - t_-)^3 & t_- < x < t_+ \\ x - t & x \geq t_+ \end{cases} \tag{18a}$$

with $t_- \leq t \leq t_+$. Setting

$$\begin{aligned} p &= (2t_+ + t_- - 3t)/(t_+ - t_-)^2 \\ r &= (2t - t_+ - t_-)/(t_+ - t_-)^3 \end{aligned} \tag{18b}$$

causes these basis functions to be continuous and have continuous first derivatives. This approximation has discontinuous second derivatives at the side knot locations, t_- and t_+. The central knot t, is placed at the corresponding knot location of $b^{(1)}(x|t)$. The two side knots, t_- and t_+, are placed at the midpoints between adjacent central knots on the same variable thereby minimizing

the number of second derivative discontinuities. The (central) knots are placed using the $b^{(1)}(x|t)$ (11) basis, taking advantage of the corresponding speed and numerical stability. The approximation with continuous derivatives is accomplished through using the corresponding piecewise cubic basis (18).

The analogue to this approach in the more general setting of MARS modeling is to perform derivative smoothing in the ANOVA representation (14). Each distinct ANOVA function (14b), (14c), (14d), etc. is smoothed separately. The side knots are placed at the midpoints between the central knot locations as projected onto each variable defining the particular function. For the additive ANOVA functions (14b) this of course reduces to the Friedman and Silverman (1987) strategy. Replacing each $b^{(1)}(x|t)$ (11) by its corresponding $C(x|t_-, t, t_+)$ (18) in the MARS model (13) (14) results in a continuous approximation with everywhere continuous derivatives.

2.4. Knot Optimization.

A natural strategy would be to make each distinct observation abscissa value on each predictor variable a potential location for knot placement. Friedman and Silverman (1987) argue that a more effective strategy is to restrict the number of candidate knot locations to very Lth (distinct) observation abscissa value, with L given by

$$L(p, N) = -\log_2 \left[-\frac{1}{pN} \ell n(1-\alpha) \right] /2.5 \tag{19}$$

and $0.05 \leq \alpha \leq 0.01$. The considerations that lead to this result do not change when one considers the more general MARS setting.

2.5. Computational Considerations.

In order for any method to be practical it must be computationally feasible. If implemented in a straightforward manner the approximation strategy we propose would require prohibitive computation. A full $M + 1$ parameter linear least squares fit for he coefficients $\{a_m\}_0^M$ must be performed to evaluate the model selection criterion (15). This must be done at every potential knot location on every variable for all M (previous) terms at each step M. The only way this can be made to be computationally feasible is through updating formulae. That is, given the solution fit at one potential knot location, the solution at the next one can be obtained through rapidly computable simple updates of the previous solution. Friedman and Silverman (1987, Section 2.3) derived updating formulae for the quantities that enter into the normal equations of the least squares fit for the additive modeling case. Analogous updating formulae can be derived for the more general case of MARS modeling. Use of these updating formulae reduce the computation from being proportional to M^4pN^2/L to M^3pN/L. As a point of reference, the computation for the three examples (Section 3) each required about two minutes on a SUN Microsystems model 3/260.

3.0. Examples.

This section provides four illustrations of MARS modeling. The data are simulated so that the results can be compared with the known (generated) truth. The first and fourth examples are purely contrived, whereas the middle two are taken form electrical engineering. In all examples the smoothing parameter d (15b) was taken to be $d = 2.5$. (The software automatically reduces it to $d_a = 0.8d = 2.0$ for additive modeling.) The minimum number of observations between knot locations was determined by (19). In all examples the explanatory variables were standardized to aid in numerical stability. (The MARS procedure is, except for numerics, invariant to the predictor variable scales.) The response variable was also standardized so that the GCV score would be an estimate for the fraction of unaccounted for variance ($e^2 = 1 - R^2$).

3.1. Simple Function of Ten Variables.

For this example, $N = 100$ covariate vectors were uniformly generated in a $n = 10$ dimensional unit hypercube. Associated with each such covariate vector is a response value generated as

$$\begin{aligned} y_i = {} & 0.02e^{4x_{1i}+3x_{2i}} + 5\sin(\pi x_{3i}/2) \\ & + 3x_{4i} + 2x_{5i} + 0 \cdot x_{6i} + 0 \cdot x_{7i} + 0 \cdot x_{8i} \\ & + 0 \cdot x_{9i} + 0 \cdot x_{10,i} + \epsilon_i, \quad 1 \leq i \leq 100, \end{aligned} \tag{20}$$

with the ϵ_i generated from a standard normal distribution. The ratio of standard deviations of the signal to the noise is 3.08 so that the true underlying function accounts for 91% of the variance of y.

The underlying function (20) consists of an interaction in the first two variables, an additive nonlinear dependence in the third, and linear dependencies in the fourth and fifth. The last five, $x_6 - x_{10}$, are pure noise variables independent of the response.

Table 1 displays the results of applying the MARS procedure to these data. Table 1a shows the history of the forward stepwise knot placement. The second column gives the GCV score (15) at each iteration M (first column). The third column shows the effective number of parameters in the fit $C(M)$ (15b). The fourth and fifth columns give the optimizing knot variable v^* and location t^*, while the last column points to the optimizing previous term (multivariate spline basis function) ℓ^* that multiplies the new univariate spline function. This term may in fact point to previous terms for its definition. The value $\ell^* = 0$ indicates that the previous multiplying term is B_0 (9b) so that a new purely additive term is being included in the model. The particular factors comprising the Mth multivariate spline basis function are identified by starting with the Mth row, then preceeding to its parent, then to its parent's parent and so on, until reaching a parent value of $\ell^* = 0$.

Table 1a shows that the first knot was placed on x_1. The second knot was placed on x_2, multiplying the first term. At this point ($M = 2$) the model consists of an additive contribution on x_1 and an interaction between x_1 and x_2. The next three iterations include purely additive contributions form x_3, x_4, and x_5. The next iteration ($M = 6$) includes an additive term in x_2. This is multiplied by a factor involving x_1 on the subsequent iteration ($M = 7$), resulting in two bivariate splines characterizing the interaction between x_1 and x_2. Up to this point the GCV score has been monotonically decreasing.

The eighth iteration places into the model a term involving an interaction between variables x_9, x_2, and x_1. Note, however, that the GCV score has increased slightly. As more terms are added, the GCV score continues to increase until the present maximum number of terms $M_{\max} = 17$, is reached.

Table 1b shows the result of the backwards stepwise term deletion strategy. The first column gives the term number, m, the second its least squares coefficient, a_m (9a), followed by the knot variable, location, and parent as in Table 1a. A zero coefficient value, $a_m = 0$, means that the term has been deleted. Note that in addition to the deletion of all terms beyond $M = 7$, the purely additive contributions of variables x_1 and x_2 (first and sixth terms) have also been deleted. This leaves only the two terms (second and seventh) involving pure interactions between these two variables.

Table 1c summarizes the ANOVA decomposition of the final model. There are four ANOVA functions. The first three are additive functions on variables x_3, x_4, and x_5 respectively. The fourth ANOVA function is bivariate and represents a (pure) interaction between x_1 and x_2. Table 1c also gives the GCV score for the fit with the corresponding piecewise cubic basis (18). It is seen to be essentially the same as for the piecewise linear basis given in Table 1b.

The second column in Table 1c gives the standard deviation of the corresponding ANOVA function. This gives one indication of its (relative) importance to the model and is interpreted in a manner similar to a (standardized) regression coefficient in a linear model. The third column gives

Table 1a

History of the MARS forward stepwise knot placement strategy for Example 3.1.

iter.	gcv	# efprms	variable	knot	parent
1	0.8460	4.5	1.	0.5257	0.
2	0.5781	8.0	2.	-0.6736	1.
3	0.3914	11.5	3.	-1.626	0.
4	0.2885	15.0	4.	-1.170	0.
5	0.2347	18.5	5.	-1.601	0.
6	0.1911	22.0	2.	-1.177	0.
7	0.1599	25.5	1.	-1.164	6.
8	0.1603	29.0	9.	-1.128	2.
9	0.1621	32.5	3.	-0.9315	0.
10	0.1696	36.0	4.	1.015	1.
11	0.1802	39.5	3.	1.013	0.
12	0.1829	43.0	6.	-0.2161	11.
13	0.1936	46.5	4.	-1.675	5.
14	0.2062	50.0	4.	0.2366e-01	11.
15	0.2271	53.5	9.	1.583	3.
16	0.2519	57.0	9.	-0.2349	5.
17	0.2837	60.5	2.	-0.4146	5.

Table 1b

The result of the backwards stepwise term deletion strategy
for Example 3.1.

$gcv = 0.1404$ $\quad \#efprms = 18.5$

term	coeff.	variable	knot	parent
1	0.	1.	0.5257	0.
2	0.8746	2.	-0.6736	1.
3	0.4525	3.	-1.626	0.
4	0.3171	4.	-1.170	0.
5	0.2232	5.	-1.601	0.
6	0.	2.	-1.177	0.
7	0.2373	1.	-1.164	6.
8	0.	9.	-1.128	2.
9	0.	3.	-0.9315	0.
10	0.	4.	1.015	1.
11	0.	3.	1.013	0.
12	0.	6.	-0.2161	11.

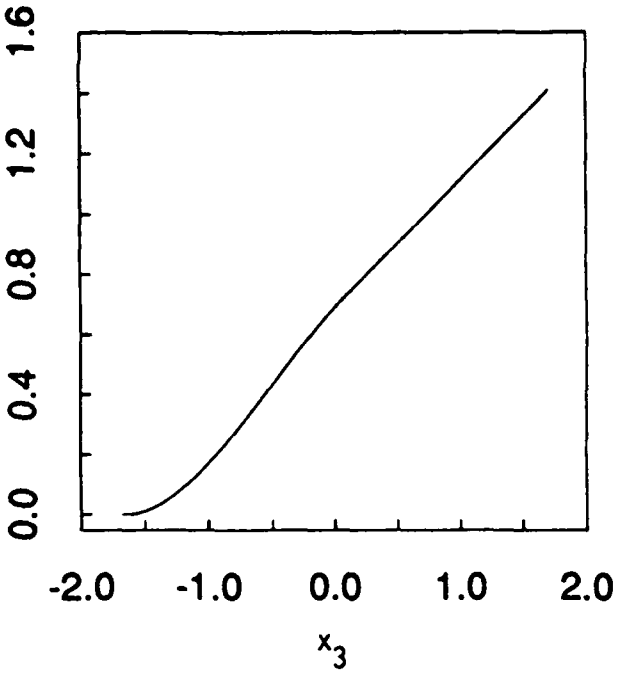
Table 1c

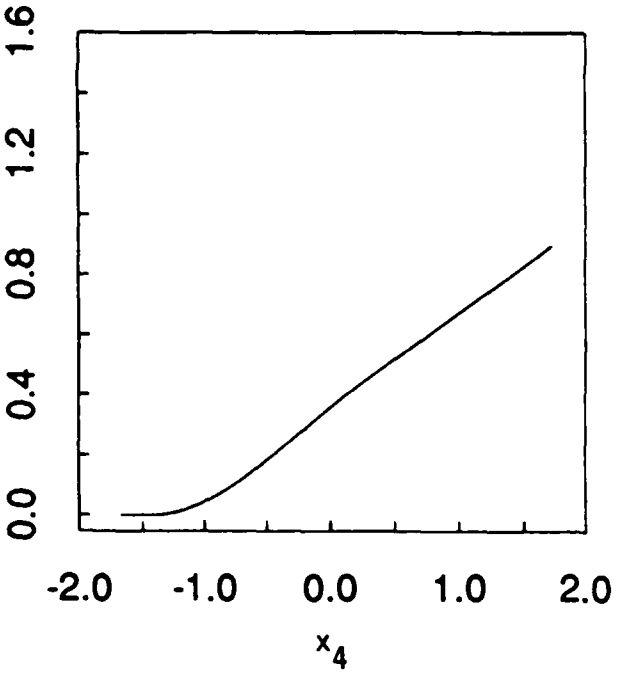
ANOVA decomposition summary of the MARS model for Example 3.1

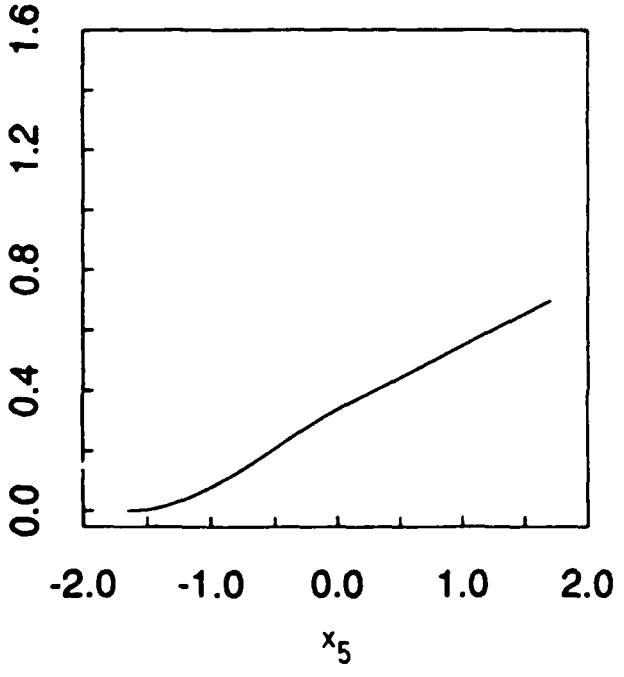
fun.	std. dev.	-gcv	# terms	# efprms	variable(s)	
1	0.4518	0.4109	1	3.5	3	
2	0.2983	0.2520	1	3.5	4	
3	0.2229	0.1974	1	3.5	5	
4	0.7772	0.8867	2	7.0	1	2

piecewise cubic fit on 5 terms, gcv = 0.1457

Figure 1a: Graphical representation of the ANOVA decomposition of the piecewise cubic MARS model for Example 3.1.







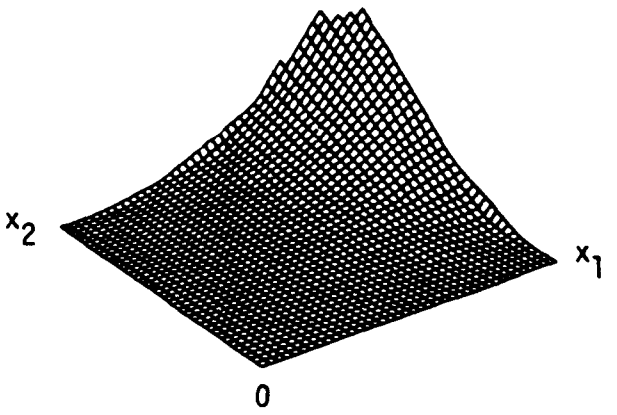


Figure 1b: Enlargement of the fourth frame of Figure 1a; interaction contribution of (x_1, x_2) to the MARS model for Example 3.1.

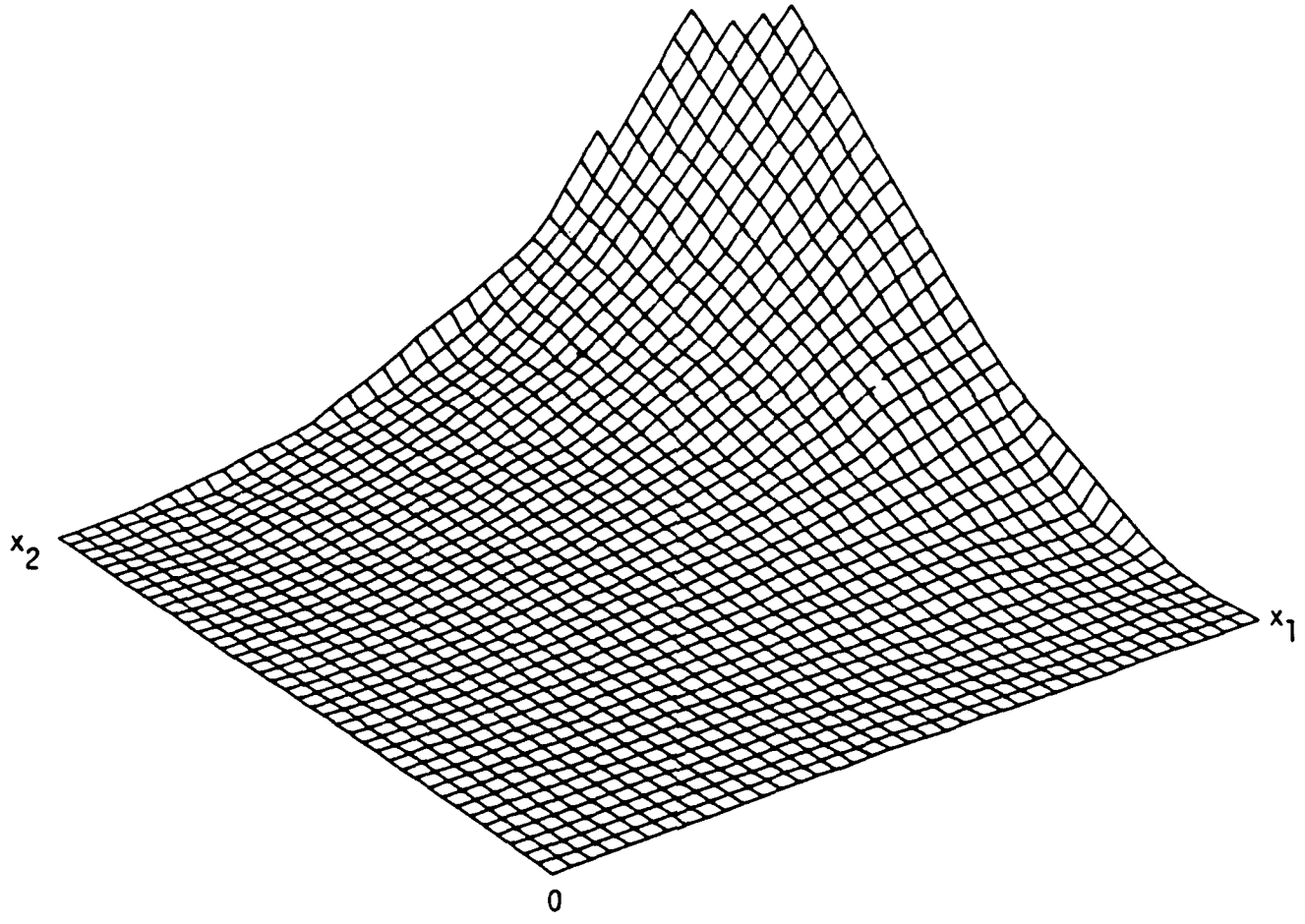
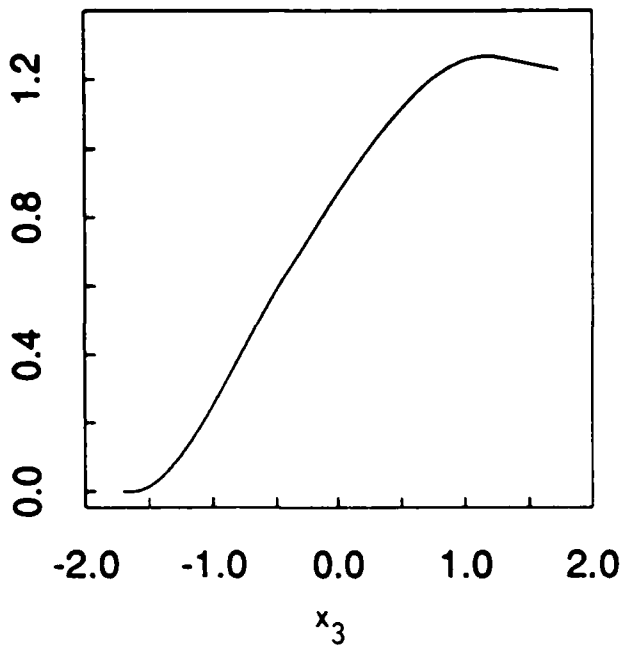
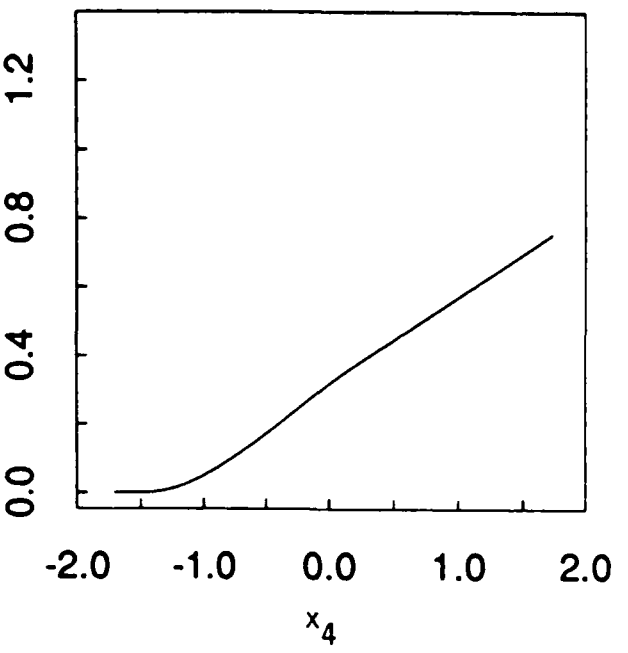
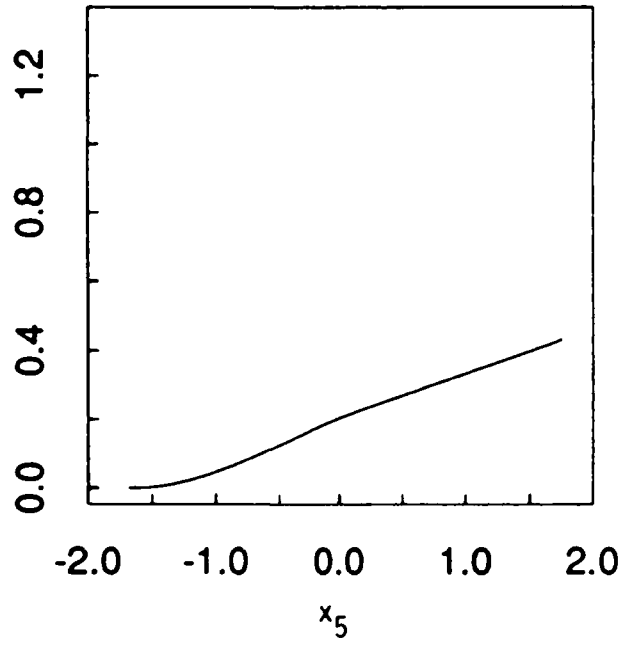
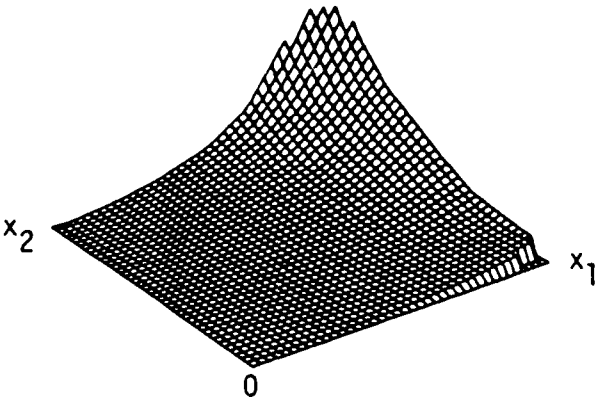


Figure 1c: Graphical ANOVA decompositon of the MARS model for Example 3.1, with 200 observations.









another indication of the importance of the corresponding ANOVA function, by providing the GCV score for the model with all of the terms corresponding to that particular ANOVA function deleted. This can be used to judge whether this ANOVA function is making an important contribution to the model, or whether it just slightly improves the global GCV score. In this example all four ANOVA functions appear to be important with the third one, involving x_5, being the weakest.

Figure 1a provides a pictorial representation of the ANOVA decomposition by plotting the respective (piecewise-cubic) ANOVA functions. The first three frames plot the respective additive functions involving x_3, x_4, and x_5. The fourth frame provides a perspective mesh plot of the bivariate ANOVA function involving x_1 and x_2. Figure 1b is an enlargement of the fourth frame of Figure 1a.

These figures show very nearly linear dependencies on x_3, x_4, and x_5, and a strong nonlinear interaction between x_1 and x_2. It is important to note that Figure 1b does not represent a smooth of the response y on variables x_1 and x_2, but rather it shows the contribution of x_1 and x_2 to a smooth of y on variables $x_1, \cdots, x_{10}$. The accuracy of the resulting approximation is fairly remarkable considering the high dimensionality, $n = 10$, and the small sample size, $N = 100$. Note also that the procedure (correctly) did not enter $x_6, \cdots, x_{10}$ into the model.

The only shortcoming of the MARS model based on these data is that it did not capture the nonlinearity in the additive contribution of x_3 (20). Figure 1c shows the pictorial representation of the ANOVA decomposition corresponding to Figure 1a when the sample size is increased to $N = 200$. The model looks very similar to that for the smaller ($N = 100$) sample size (Figure 1a) except that it now gives a better approximation to the contribution of x_3.

Tables 1a – 1c and Figures 1a – 1b illustrate the application of the MARS procedure to a single data set (replication) from the particular setting under study (20). They do not give information on the average performance of the procedure when applied to this situation. Table 1d displays the results of a simulation study that addresses this issue. Each row summarizes the results of 100 replications of the following procedure. A sample of N ten-dimensional covariate vectors were randomly sampled from a uniform distribution in $[0, 1]^{10}$. A sample of N random standard normal deviates were then generated and the corresponding response values (20) were assigned to the covariate vectors. The MARS procedure was then applied. A new data set of 5000 observations was then generated and used to estimate the normalized integrated squared error

$$ISE = \int [f(\mathbf{x}) - \hat{f}(\mathbf{x})]^2 d^{10}x / \mathrm{Var}_{\mathbf{x}} f(\mathbf{x}), \tag{21a}$$

and the normalized predictive squared error

$$PSE = (ISE \cdot \mathrm{Var}_{\mathbf{x}} f(\mathbf{x}) + 1)/(\mathrm{Var}_{\mathbf{x}} f(\mathbf{x}) + 1) \tag{21b}$$

(fraction of unaccounted for variance) for the piecewise cubic MARS model.

The second column of Table 1d gives the optimizing GCV score averaged over the 100 replications, whereas the third and fourth columns give the corresponding average PSE and ISE (21) respectively. The quantities in parentheses are the associated standard deviations over the 100 replications. (The standard deviations of the averages are one tenth these values.)

Table 1d shows results for three sample sizes ($N = 50, 100, 200$) and for three sets of constraints applied to the MARS model. These constraints involve the maximum number of factors mi that are permitted to enter a single multivariate spline basis function. This controls the maximum interaction order permitted in the model. Setting $mi = 1$ restricts the model to be additive in the predictor variables, whereas $mi = 2$ limits the model to interactions involving at most two variables, and so on. The value $mi = n$ results in no restriction. Limiting the interaction level of the MARS model can improve accuracy (reduce variance) if the true underlying function f is close

Table 1d

Summary of 100 replications of Example 3.1, piecewise cubic fit.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 50$:			
1	.46 (.12)	.45 (.097)	.40 (.11)
2	.28 (.13)	.28 (.18)	.22 (.20)
10	.27 (.11)	.30 (.19)	.24 (.21)
$N = 100$:			
1	.36 (.072)	.36 (.064)	.30 (.070)
2	.15 (.043)	.14 (.026)	.059 (.029)
10	.15 (.047)	.16 (.041)	.077 (.044)
$N = 200$:			
1	.32 (.037)	.31 (.022)	.25 (.023)
2	.12 (.029)	.12 (.015)	.033 (.015)
10	.12 (.029)	.12 (.024)	.041 (.025)

Table 2a

ANOVA decomposition summary of the MARS model
on alternating current series circuit impedence, Z.

$gcv = 0.2311$ $\quad \#efprms = 46.5$

fun.	std. dev.	-gcv	# terms	# efprms	variable(s)	
1	0.5096	0.6392	1	3.5	1	
2	1.833	0.6854	3	10.5	2	
3	1.417	0.6431	3	10.5	4	
4	0.4195	0.4401	1	3.5	2	3
5	2.034	0.5704	4	14.0	2	4
6	0.1702	0.2577	1	3.5	3	4

piecewise cubic fit on 13 terms, gcv =0.2447

to an $\hat{f}$ that involves at most low order interactions. If not, such a limitation will introduce some bias in exchange for the corresponding variance reduction. In terms of interpretability there is a strong advantage to models with $mi = 2$, owing to their graphical representation by means of the ANOVA decomposition.

In terms of ISE (21a) the accuracy of the MARS model for this problem is seen to increase rapidly as the sample size increases from 50 to 200. The additive model ($mi = 1$) is seen to be distinctly inferior to those involving interactions ($mi = 2, 10$) especially as the sample size increases. The optimizing GCV score is seen very slightly to overestimate the true PSE on average.

The true underlying function (20) in this case happens to involve at most interactions in two variables. Thus, setting $mi = 2$ results here in no increase in bias. Owing to the decrease in variance, the ISE is seen to be somewhat better than for the unrestricted MARS model ($mi = 10$). The size of the effect is seen, however, to be fairly small ($\leq 25\%$ in squared error loss) so that a large penalty is not incurred by fitting the full nonparametric model.

3.2. Alternating Current Series Circuit.

Figure 2a shows a schematic diagram of a simple alternating current series circuit involving a resistor R, inductor L, and capacitor C. Also in the circuit is a generator that places a voltage

$$V_{ab} = V_o \sin \omega t \tag{21a}$$

across the terminals a and b. Here ω is the angular frequency which is related to the cyclic frequency f by

$$\omega = 2\pi f. \tag{21b}$$

The electric current I_{ab} that flows through the circuit is also sinusoidal with the same frequency,

$$I_{ab} = (V_o/Z)\sin(\omega t - \phi). \tag{21c}$$

Its amplitude is governed by the impedance Z of the circuit and there is a phase shift ϕ, both depending on the components in the circuit:

$$Z = Z(R, \omega, L, C),$$
$$\phi = \phi(R, \omega, L, C).$$

From elementary physics one knows that

$$Z(R, \omega, L, C) = [R^2 + (\omega L - 1/\omega C)^2]^{1/2}, \tag{22a}$$

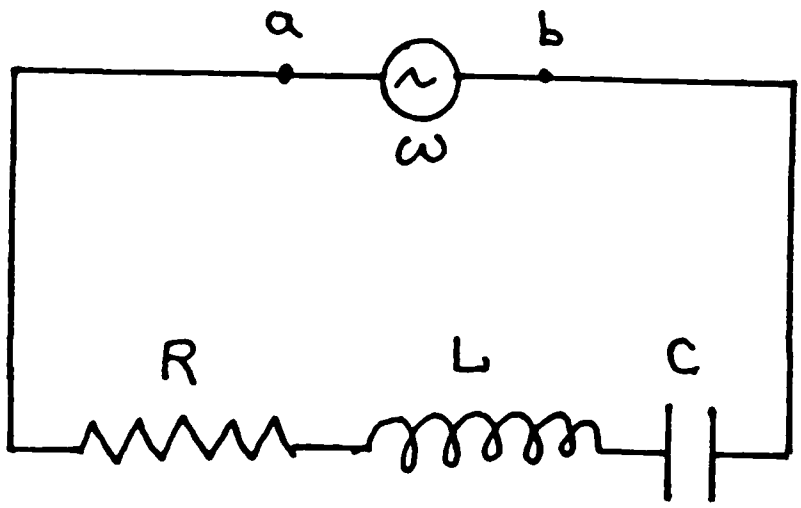
$$\phi(R, \omega, L, C) = \tan^{-1}\left[\frac{\omega L - 1/\omega C}{R}\right] \tag{22b}$$

The purpose of this exercise is to see to what extent the MARS procedure can approximate these functions and perhaps yield some insight into the variable relationships, in the range

$$\begin{aligned} x_1&: \ 0 \leq R \leq 100 \ \text{ohms} \\ x_2&: \ 20 \leq f \leq 280 \ \text{hertz} \\ x_3&: \ 0 \leq L \leq 1 \ \text{henries} \\ x_4&: \ 1 \leq C \leq 11 \ \text{micro farads.} \end{aligned} \tag{23}$$

Two hundred four-dimensional uniform covariate vectors were generated in the ranges (23). For each one, two responses were generated by adding normal noise to (22a) and (22b). The variance

Figure 2a: Schematic diagram of the alternating current series circuit of Example 3.2.



of the noise was chosen to give a 3 to 1 signal to noise ratio for both Z (22a) and ϕ (22b), thereby causing the true underlying function to account for 90% of the variance in both cases.

3.2.1. Impedance, Z.

Applying the MARS procedure to the impedance data with $mi = 1$ (additive model) gave an optimizing GCV score of 0.558. The GCV scores for $mi = 2$ and 4 were respectively 0.231 and 0.229. The additive model is seen (not surprisingly) to be inadequate. Perhaps more surprising is the fact that even though the true underlying function (22a) contains interactions to all orders, an approximation involving only two-variable interactions is seen to give nearly as good a fit to these data. Owing to its increased interpretability we show the results of the $mi = 2$ model.

Table 2a shows the ANOVA decomposition in the same format as Table 1c. There is a purely additive contribution from $x_1(R)$, additive contributions from $x_2(\omega)$ and $x_4(C)$, and interactions amongst x_2, $x_3(L)$, and x_4. Of the six ANOVA functions, all but the last one (involving an interaction between the capacitance C and the inductance L) seem important to the model. Figure 2b displays a graphical representation of the ANOVA decomposition. The first frame plots the (additive) contribution from the resistance R. The next three frames display the contributions of the remaining variables that participate in interactions. These perspective mesh plots show the total (additive plus interaction) contributions of each such variable pair. For example, the frame in the upper right corner plots the sum of the second and fourth ANOVA functions, whereas that of the lower left plots the sum of the second, third, and fifth.

The plots have been rotated so as to provide the best perspective view. The indicated zero marks the lowest value and the axis label marks the direction of higher values.

The dependence of the impedance Z on R (first frame) is estimated to be approximately linear. For low frequencies ω, Z is seen to be high and independent of L (upper right frame). For high ω, Z has a mild monotonically increasing dependence on L. For low L, Z monotonically decreases with increasing ω, whereas for high L values, the impedance is seen to achieve a minimum for moderate ω values. The lower left frame shows that Z is very small and roughly independent of ω and C except when they jointly have very small values, in which case the impedance increases dramatically. The lower right frame of Figure 2b shows that the C, L joint contribution is nearly additive, consistent with the weak contribution of the sixth ANOVA function (Table 2a) to the MARS model.

These interpretations are based on visual examination of the graphic representation of the ANOVA decomposition of the MARS approximation, based on a sample of size $N = 200$. Since the data in this case are generated from known truth one can examine the generating equation (22a) to verify their general correctness.

Table 2b summarizes the results of a simulation study based on 100 replications of data randomly drawn according to the above prescription (22a), (23), in the same format as Table 1d. The MARS procedure applied to the smallest sample size, $N = 100$, is seen to provide a fairly poor approximation on average in terms of ISE. The approximation accuracy improves substantially with the larger samples, except for additive modeling ($mi = 1$). The approximation accuracy for the constrained ($mi = 2$) models is (on average) nearly identical to the unconstrained ($mi = 4$) ones. It appears that the bias-variance trade-off is exactly off-setting in this case.

The average GCV score is seen to underestimate the corresponding PSE at the smallest sample size. This is due to the sharp joint dependence of Z on ω and C [see (22a) and Figure 2, third frame]. For small sample sizes most replications will fail to sample covariate vectors with very small joint values for ω and C, thereby failing to capture the rapid variation of Z in that region. There is no way that the GCV score (based on the ASR) can detect rapid function variation where there is no data. Note that sample reuse techniques such as cross-validation or bootstrapping have the same problem. As the sample size increases enough data is sampled in this region and the GCV score gives a more accurate estimate of the true PSE (on average).

3.22. Phase Angle, ϕ.

Figure 2b: Graphical ANOVA decomposition for the alternating current series circuit impedance, Z, Example 3.21.

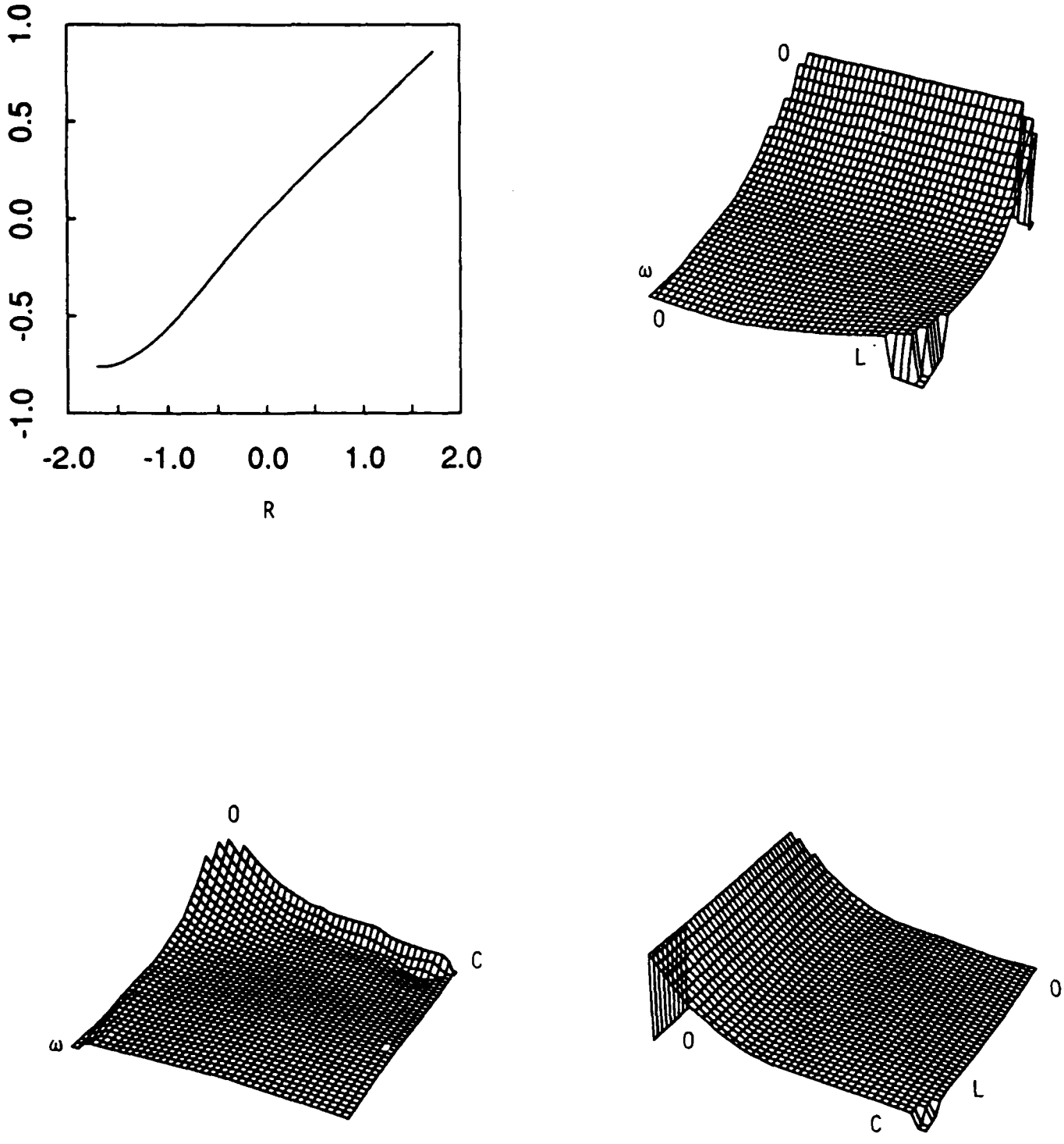


Table 2b

Summary of 100 replications of the alternating current series circuit impedance, Z, piecewise cubic fit.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 100$:			
1	.65 (.12)	.71 (.092)	.68 (.10)
2	.46 (.15)	.52 (.19)	.46 (.21)
4	.45 (.15)	.52 (.19)	.47 (.21)
$N = 200$:			
1	.60 (.082)	.62 (.050)	.58 (.056)
2	.27 (.064)	.27 (.10)	.20 (.11)
4	.28 (.066)	.28 (.091)	.20 (.11)
$N = 400$:			
1	.57 (.049)	.57 (.026)	.52 (.029)
2	.20 (.057)	.18 (.050)	.095 (.056)
4	.20 (.035)	.18 (.035)	.092 (.038)

Table 3a

ANOVA decomposition of the MARS model on the alternating current series circuit phase angle, ϕ.

$gcv = 0.2190$ $\quad \#efprms = 39.5$

fun.	std. dev.	-gcv	# terms	# efprms	variable(s)	
1	0.6323	0.3257	1	3.5	2	
2	0.7253	0.4180	2	7.0	4	
3	0.9931	0.3041	1	3.5	1	
4	0.6483	0.4015	2	7.0	2	3
5	0.1521	0.2254	1	3.5	2	4
6	0.7754	0.2662	2	7.0	1	4
7	0.2064	0.2248	1	3.5	1	3
8	0.3464	0.2458	1	3.5	1	2

piecewise cubic fit on 11 terms, gcv = 0.2393

The MARS procedure applied to the phase angle data (22b) (23) with $mi = 1, 2$, and 4 gave optimizing GCV scores of 0.295, 0.219, and 0.203, respectively. Here the additive model, while still being less accurate, is more competitive with those involving interactions. The two variable interaction model again fits the data almost as well as the unconstrained model.

Table 3a summarizes the ANOVA decomposition for the $mi = 2$ MARS model. It involves additive contributions from all but $x_3(L)$ and interactions among all variable pairs except C and L. Two of the ANOVA functions (fifth and seventh) however are seen to make very weak contributions to the final model. Figure 2c is a graphical representation of the ANOVA decomposition in the same format as Figure 2b. The dependence of the phase angle ϕ on all of the variables is seen to be more gentle and more nearly additive than the impedance Z (Figure 2b). The principal interaction effect is to decrease the phase angle for simultaneously high values of the predictor variable pairs.

Table 3b gives the results of 100 replications of phase angle data generated according to (22b), (23). At the smallest sample size ($N = 100$) the additive model produces fits that (on average) are nearly as accurate as those involving interactions. For the larger samples the interaction models are somewhat more accurate in terms of ISR. The average optimizing GCV score is seen to be quite close to the true average PSE.

3.3. Additive Data.

In the preceding examples there were strong interaction effects and it was seen that allowing such effects in the MARS model substantially improved approximation accuracy. This example, taken from Friedman and Silverman (1987), examines what happens when the true underlying function is exactly additive and interactions are allowed to enter the MARS model. One would expect accuracy to deteriorate since allowing for interactions among the variables increases the variance of $\hat{f}$ while, in this particular case, not decreasing the bias.

Table 4 summarizes (in the same format as Tables 1d, 2b, 3b) the results of 100 replications of the following simulation experiment. $N(= 50, 100, 200)$ 10-dimensional covariate vectors were generated in the unit hypercube. A set of standard normal deviates ϵ_i were then generated and response values were assigned according to

$$\begin{aligned} y_i = {} & 0.1e^{4x_{1i}} + 4/[1 + e^{-20(x_{2i}-1/2)}] \\ & + 3x_{3i} + 2x_{4i} + x_{5i} + 0 \cdot x_{6i} + 0 \cdot x_{7i} \\ & + 0 \cdot x_{8i} + 0 \cdot x_{9i} + 0 \cdot x_{10,i} + \epsilon_i. \end{aligned}$$

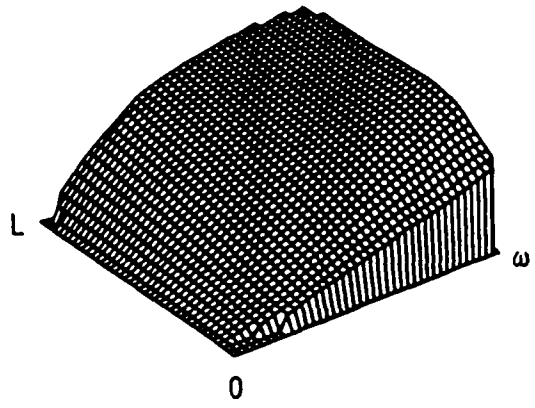
Here the signal to noise ratio is 0.28 so that the true underlying function accounts for 92% of the variance of the response.

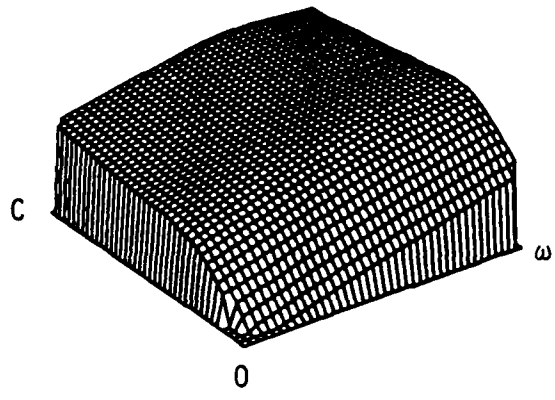
The ratio of the average ISE values for the additive and $mi = 2$ interaction fits are seen (Table 4) to be about 0.67 at all sample sizes. The corresponding ratio for the $mi = 10$ unconstrained fit is about 0.60. The corresponding square roots of the ratios are 0.81 and 0.77. Thus, the (average) accuracy here is reduced by about 25% when the interactive models are fit to purely additive data. This degradation is surprisingly small given the small sample sizes and the high dimensionality ($n = 10$). Note that the average GCV scores for the interactive models are always slightly worse than that for the corresponding additive fit, so that the interactive models are not (on average) claiming to do better than the additive ones. This suggests a strategy of accepting the additive model if those involving interactions fit no better in terms of the GCV score, especially owing to the increased interpretability of the additive model.

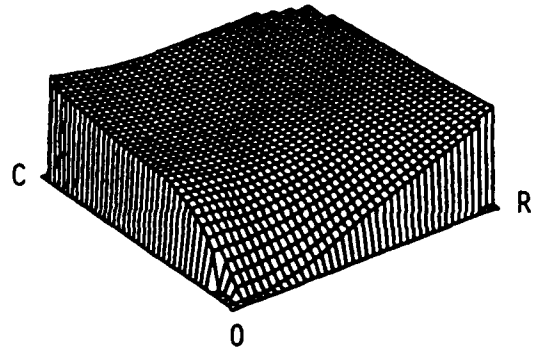
4.0. Remarks.

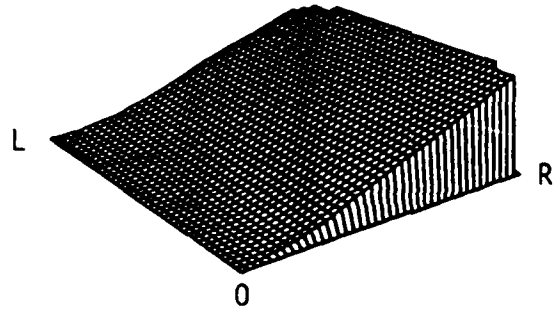
This section covers various aspects (extensions, limitations, etc.) of the MARS procedure not discussed in the previous sections.

Figure 2c: Graphical ANOVA decomposition for the alternating current series circuit phase angle, ϕ, Example 3.22.









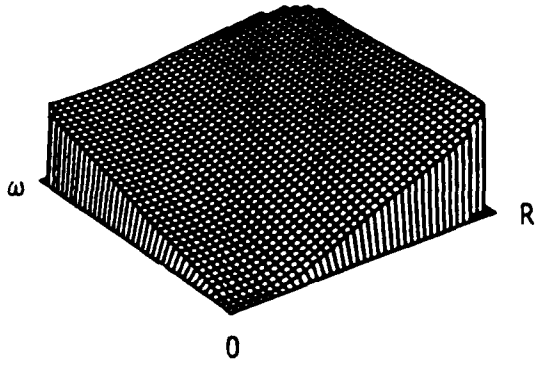


Table 3b

Summary of 100 replications of the alternating current series circuit phase angle, ϕ, piecewise cubic fit.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 100$:			
1	.36 (.057)	.35 (.036)	.27 (.040)
2	.33 (.059)	.32 (.047)	.25 (.052)
4	.32 (.059)	.33 (.12)	.26 (.14)
$N = 200$:			
1	.32 (.032)	.31 (.016)	.23 (.017)
2	.25 (.033)	.24 (.022)	.15 (.025)
4	.24 (.032)	.24 (.022)	.15 (.070)
$N = 400$:			
1	.30 (.020)	.29 (.007)	.21 (.008)
2	.22 (.019)	.20 (.011)	.11 (.012)
4	.21 (.019)	.19 (.012)	.10 (.013)

Table 4

Summary of 100 replications of applying MARS to purely additive data, Example 3.3.

mi	$\overline{GCV}$	$\overline{PSE}$	$\overline{ISE}$
$N = 50$:			
1	.30 (.092)	.25 (.053)	.13 (.062)
2	.34 (.077)	.30 (.074)	.19 (.085)
10	.34 (.077)	.29 (.080)	.19 (.092)
$N = 100$:			
1	.22 (.035)	.18 (.020)	.053 (.024)
2	.22 (.040)	.21 (.035)	.081 (.041)
10	.24 (.041)	.21 (.035)	.088 (.042)
$N = 200$:			
1	.17 (.022)	.16 (.008)	.024 (.009)
2	.18 (.024)	.17 (.014)	.036 (.016)
10	.19 (.025)	.17 (.012)	.040 (.015)

4.1. Constraints.

The MARS procedure is nonparametric in that it attempts to model arbitrary functions. It is often appropriate, however, to place constraints on the final model, dictated by knowledge of the system under study, outside the specific data at hand. Such constraints will reduce the variance of the model estimates, and if the outside knowledge is fairly accurate, not substantially increase the bias. One type of constraint has already been discussed in Section 3, namely limiting the maximum interaction order of the model. One might in addition (or instead) limit the specific variables that can participate in interactions. If it is known a priori that certain variables are not likely to interact with others, then restricting their contributions to be at most additive can improve accuracy. If one further suspects that specific variables can only enter linearly, then placing such a restriction can improve accuracy. The incremental charge d (15b) for knots placed under these restrictions should be less than that for the unrestricted knot optimization. (The implementing software charges $0.8 \cdot$ df and $0.4 \cdot$ df, respectively, for the additive and linear constraints where df is the charge for unrestricted knot optimization.)

These constraints, as well as far more sophisticated ones, are easily incorporated in the MARS strategy. Before each prospective knot is considered, the parameters of the corresponding potential new multivariate spline basis function (v, t, ℓ, and B_ℓ) (10) can be examined for consistency with the constraints. If it is inconsistent, it can simply be marked ineligible for inclusion in the model.

4.2. Semiparametric Modeling.

Another kind of a priori knowledge that is sometimes available has to do with the nature of the dependence of the response on some (or all) the predictor variables. The user may be able to provide a function $g(x_1, \cdots, x_n)$ that is thought to capture some aspects of the true underlying function $f(x_1, \cdots, x_n)$. More generally, one may have a set of such functions $\{g_j(x_1, \cdots, x_n)\}_1^J$, each one of which might capture some aspect of the functional relationship. A semiparametric model of the form

$$\hat{f}_{sp}(x_1, \cdots, x_n) = \sum_{j=1}^{J} c_j g_j(x_1, \cdots, x_n) + \hat{f}(x_1, \cdots, x_n), \tag{24}$$

where $\hat{f}(x_1, \cdots, x_n)$ takes the form of the MARS approximation (9), could then be fit to the data. The coefficients c_j in (24) are jointly fit along with the parameters of the MARS model. To the extent that one or more of the g_j successfully describe attributes of the true underlying function, they will be included with relatively large (absolute) coefficients, and the accuracy of the resulting (combined) model will be improved.

Semiparametric models of this type (24) are easily fit using the MARS strategy. One simply includes $\{g_j(x_1, \cdots, x_n)\}_1^J$ as J additional predictor variables $(x_{n+1}, \cdots, x_{n+J})$ and constrains their contributions to be linear. One could also, of course, not place this constraint, thereby fitting more complex semiparametric models than (24).

4.3. Collinearity.

Extreme collinearity of the predictor variables is a fundamental problem in the modeling of observational data. Solely in term of predictive modeling it represents an advantage in that it effectively reduces the dimensionality of the predictor variable space. This is provided that the observed collinearity is a property of the population distribution and not an artifact of the sample at hand. Collinearity presents, on the other hand, severe problems for interpreting the resulting model.

This problem is even more serious for (interactive) MARS modeling than for additive or linear modeling. Not only is it difficult to isolate the separate contributions of highly collinear predictor variables to the functional dependence, it is difficult to separate additive and interactive contribu-

tions among them. A highly nonlinear dependence on one such variable can be well approximated by a combination of functions of several of them, and/or by interactions among them.

In the context of MARS modeling one strategy to cope with this (added) problem is to fit a sequence of models with increasing maximum interaction order (mi). One first fits an additive model ($mi = 1$), then one that permits at most two variable interactions ($mi = 2$), and so on. The models in this sequence can then be compared by means of their respective optimizing GCV scores. The one with the lowest mi value that gives a (relatively) acceptable fit can then be chosen.

4.4. Robustness.

Since the MARS method as described here uses a model selection criterion based on squared error loss it is not robust against outlying response values. Unlike linear regression, however, it is not very sensitive to outliers in the predictor variable space, owing to the local nature of the resulting fit; sample covariate vectors far from an evaluation point tend to have less rather than more influence on the model estimate. Response outliers will tend to strongly effect model estimates only close to their corresponding covariate values. They will also (slightly) increase the variance of model estimates elsewhere by increasing the number of multivariate spline basis functions (required to capture the apparent high curvature of the function near each outlier).

There is nothing fundamental about squared-error loss in the MARS approach. Any criterion can be used to select the multivariate spline basis functions, and construct the final fit, by simply replacing the internal linear least squares fitting routine by one that minimizes another loss criterion (given the current set of multivariate spline basis functions). Using robust/resistant regression methods would provide resistance to outliers.

The only advantage to squared-error loss in the MARS context is computational. It is difficult to see how rapid updating formulae could be developed for other types of linear fitting. For those with access to rich computing environments, this presents no problem. For others, a compromise strategy can mitigate the robustness problem for isolated outliers. The multivariate spline basis functions are selected using the standard MARS approach with least-squares fitting. Given this basis, the expansion coefficients $\{a_m\}_0^M$ (9) are then fit using a robust/resistant linear regression method to form the final model. This reduces the influence of the response outliers on model predictions close to their corresponding covariate vectors. It does not remove the (small) increased variance associated with the additional (now redundant) basis functions.

4.5. Logistic Regression.

Linear logistic regression (Cox, 1970) is often used when the response variable assumes only two values. The model takes the form

$$\log[p/(1-p)] = \sum_{i=1}^{n} \beta_i x_i$$

where p is the probability that y assumes its larger value. The coefficients $\{\beta_i\}_1^n$ are estimated by (numerically) maximizing the likelihood of the data. Recently, Hastie and Tibshirani (1986) extended this approach to additive logistic regression

$$\log[p/(1-p)] = \sum_{i=1}^{n} f_i(x_i).$$

The smooth covariate functions are estimated through their "local scoring" algorithm. The model can be further generalized by

$$\log[p/(1-p)] = \hat{f}(x_1, \cdots, x_p)$$

with $\hat{f}(x_1, \cdots, x_p)$ taking the form of the MARS approximation (9). This is implemented in the MARS algorithm by simply replacing the internal linear least-squares routine by one that does linear logistic regression (given the current set of multivariate spline basis functions). Unless rapid

updating formulae can be derived this is likely to be quite computationally intensive. A compromise strategy analogous to that described in Section 4.4, however, is likely to provide a good approximation; the multivariate spline basis functions are selected using the squared-error based loss criterion and the coefficients $\{a_m\}_0^M$ for the final model are fit using a linear logistic regression on this basis set. Note that in this setting the least-squares criterion is *more* robust than the likelihood based criterion.

4.6. Reflection Invariance.

The MARS procedure as described here is not necessarily invariant to reflections of the individual predictor variables. Replacing x_i by $-x_i$ can (slightly) change the MARS model. This is due to the fact that the pure linear term, associated with the piecewise-linear basis on each variable, is not automatically included in the model; but rather it is subjected to the same forward/backward stepwise selection strategy as all other potential basis functions. This gives the procedure the ability to model certain types of dependencies with fewer basis functions than would otherwise be the case. Also, certain kinds of interaction effects require less terms to model than others.

In order to get an idea of the size of this effect a further simulation study was performed on the alternating current series circuit example (Section 3). Fifteen additional simulation studies (N = 200, 100 replications each) were done analogous to those that led to Tables 2b and 3b. For each of the (total) 16 studies, the predictor variables were each multiplied by one of the 16 combinations of $(\pm 1, \pm 1, \pm 1, \pm 1)$. The variance of the ISE over these 16 experiments was compared to its average variance over the 100 replications of different training sample sets. For the impedance, this ratio was 0.156 whereas for the phase angle it was 0.036. The higher value for the impedance is due to the very sharp structure for very low joint values of ω and C (Figure 2, lower left frame). In both cases, however, the variability in modeling accuracy due to reflections of the predictor variables is seen to be very small compared to the variability associated with the random nature of the training data.

Several modifications of the MARS procedure that render it invariant under variable reflection are currently under study. It remains to be seen whether they can provide approximations that are as accurate as the method described here.

4.7. Low Dimensional Modeling.

The main advantage of MARS modeling over existing methodology is clearly realized in high dimensional settings. It can, however, be competitive in low dimensions ($n \leq 2$) as well. Friedman and Silverman (1987) studied its properties for the smoothing problem ($n = 1$) and showed that it can produce superior performance, especially in situations involving small samples and low signal to noise. These properties should extend to surface modeling ($n = 2$) as well, although detailed studies have not yet been performed. Friedman and Silverman (1987) also studied this approach in the special case of additive modeling ($mi = 1$). The method was shown to be competitive with existing methodology in this application, again exhibiting superior performance in situations with small samples and low signal to noise.

5.0. Conclusion.

The examples and simulation studies indicate that the MARS approach has the potential to become a useful tool for data modeling. It possesses to some degree the the desirable properties of the recursive partitioning approach; these are its adaptability, automatic variable subset selection, and ability to exploit low "local" dimensionality. Moreover, it is able to overcome some of recursive partitioning's limitations; it produces continuous approximations with continuous derivatives (if desired); it has additional adaptabilty to exploit functions with weak high order interactions, thereby providing better approximations to functions that are nearly linear or additive; and it has increased interpretability through its ANOVA decomposition that breaks up the approximation into its additive and various interaction components.

It is important to note that this is a new methodology for which there is, at present, very little collective experience. Its results should be interpreted with some caution until their reliability is tested over time in a wide variety of settings. No doubt as such experience is gained useful and important modifications to this basic approach will become apparent.

A FORTRAN program implementing the MARS methodology described in ths report is available from the author.

Bibliography

Bellman, R. E. (1961). *Adaptive Control Processes.* Princeton University Press, Princeton, New Jersey.

Bozzini, M. and Lenarduzzi, L. (1985). Local smoothing for scattered noisy data. *International Series of Numerical Mathematics* **75**, Birkhauser Verlag, Basel, 51–60.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation (with discussion). *J. Amer. Statist. Assoc.* **80**, 580–619.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth, Belmont, CA.

Breiman, L. and Meisel, W. S. (1976). General estimates of the intrinsic variability of data in nonlinear regression models. *J. Amer. Statist. Assoc.* **71**, 301–307.

deBoor, C. (1978). *A Practical Guide to Splines.* Springer-Verlag, New York, NY.

Cox, D. R. (1970). *Analysis of Binary Data*, London: Chapman and Hall.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik* **31**, 317–403.

Diaconis, P. and Shahshahani, M. (1984). On non-linear functions of linear combinations. *SIAM J. Sci. Stat. Comput.* **5**, 175–191.

Donoho, D. L. and Johnstone, I. (1985). Projection-based smoothing, and a duality with kernel methods. Department of Statistics, Stanford University, Technical Report No. 238.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *J. Amer. Statist. Assoc.* **78**, 316–331.

Friedman, J. H. (1979). A tree-structured approach to nonparametric multiple regression, in *Smoothing Techniques for Curve Estimation*, T. H. Gasser and M. Rosenblatt (eds.), Springer-Verlag, New York, 5–22.

Friedman, J. H. (1985). Classification and multiple response regression through projection pursuit, Department of Statistics, Stanford University, Technical Report LCS012.

Friedman, J. H., Grosse, E., and Stuetzle, W. (1983). Multidimensional additive spline approximation. *SIAM J. Sci. Stat. Comput.* **4**, 291–301.

Friedman, J. H. and Silverman, B. W. (1987). Flexible parsimonious smoothing and additive modeling. Stanford Linear Accelerator, Stanford, CA report SLAC-PUB-4390.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression, *J. Amer. Statist. Assoc.* **76**, 817–823.

Friedman, J. H. and Wright, M. J. (1981). A nested partitioning algorithm for numerical multiple integration. *ACM Trans. Math. Software*, March.

Hastie, T. and Tibshirani, R. (1985). Discussion of P. Huber: Projection pursuit, *Ann. Statist.* **13**, 502–508.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models (with discussion), *Statist. Science* **1**, 297–318.

Huber, P. J. (1985). Projection Pursuit (with discussion), *Ann. Statist.* **13**, 435–475.

Lyness, J. N. (1970). Algorithm 379-SQUANK (Simson Quadrature Used Adaptively – Noise Killed), *Comm. Assoc. Comp. Mach.* **13**, 260–263.

Morgan, J. N., and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal, *J. Amer. Statist. Assoc.* **58**, 415–434.

Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statist.* **33**, 1065–1076.

Shepard, D. (1964). A two-dimensional interpolation function for irregularly spaced data, *Proc. 1964 ACM Nat. Conf.*, 517–524.

Shumaker, L. L. (1976). Fitting surfaces to scattered data, in *Approximation Theory III*, G. G. Lorentz, C. K. Chui, and L. L. Shumaker, eds. Academic Press, New York, 203–268.

Shumaker, L. L. (1984). On spaces of piecewise polynomials in two variables, in *Approximation Theory and Spline Functions*, S. P. Singh et al. (eds.). D. Reidel Publishing Co., 151–197.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. Roy. Statist. Soc. B* **47**, 1–52.

Stone, C. J. (1977). Nonparametric regression and its applications (with discussion), *Ann. Statist.* **5**, 595–645.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictors (with discussion). *J. R. Statist. Soc.*, **B36**, 111–147.

III. COMPUTATIONALLY INTENSIVE STATISTICAL METHODS

Computational Aspects of Bayesian Methods
A.F.M. Smith, University of Nottingham, U.K.

A Bayesian Approach to the Design and Analysis of Computational Experiments
Toby J. Mitchell, Max D. Morris, Oak Ridge National Laboratory

Additive Principal Components: A Method for Estimating Additive Equations with Small Variance
Deborah J. Donnell, Bellcore

Stochastic Tests of Fit
P. Warwick Millar, University of California, Berkeley

Bootstrap Inference for Replicated Experiments
Walter Liggett, National Bureau of Standards

Regression Strategies
David Brownstone, University of California, Irvine

Data Sensitivity Computation for Maximum Likelihood Estimation
Daniel C. Chin, Johns Hopkins University Applied Physics Laboratory

Bootstrap Procedures in Random Effect Models for Comparing Response Rates in Multi-Center Clinical Trials
Michael F. Miller, Hoechst-Roussel Pharmaceuticals, Inc.

Bootstrapping the Mixed Regression Model with Reference to the Capital and Energy Complementarity Debate
Baldev Raj, Wilfrid Laurier University

COMPUTATIONAL ASPECTS OF BAYESIAN METHODS

A. F. M. Smith, University of Nottingham

Given a likelihood $l(x;\theta)$ and prior density $p(\theta)$, where x and θ (both typically vector-valued) denote data and unknown parameters, respectively, the starting point for Bayesian inferences about θ is the joint posterior density for θ given by

$$p(\theta|x) = \frac{l(x;\theta)p(\theta)}{\int l(x;\theta)p(\theta)\,d\theta}. \tag{1}$$

In fact, of course, we are usually interested in summaries of the full joint posterior distribution. For example, attention may be focussed on univariate marginal densities for some or all of the components θ_i of θ; bivariate joint marginal densities for various pairs (θ_i, θ_j) of component parameters; or even on simpler summaries in the form of posterior first and second moments. Alternatively, we may be interested in posterior summaries for functions of one or more of the components of θ: for example, marginal and joint densities for θ_i/θ_j and $\theta_i\theta_j$.

In all these cases, the technical key to the implementation of the formal solution given by Bayes' theorem, for specified likelihood and prior, is the ability to perform a number of integrations. First, we need to evaluate the denominator of (1) in order to obtain the normalizing constant of the posterior density; then we need to integrate over complementary components of θ, or transformations of θ, in order to obtain marginal (univariate or bivariate) densities, together with summary moments, highest posterior density intervals and regions, or whatever. Except in certain rather stylized problems (for example, exponential families together with conjugate priors), the required integrations will not be feasible analytically and so efficient numerical strategies will be required. Finally, the finite sets of numerical values obtained after marginalization need to be reconstructed into a graphical representation of a univariate or bivariate marginal posterior distribution.

We shall outline numerical integration strategies which have proved efficient and reliable for problems of this kind. A brief account will also be given of the techniques used to produce univariate density curves and contour representations of bivariate densities. Throughout, we shall provide diagramatic illustration of the main ideas.

General accounts of approaches to implementing the Bayesian paradigm are given in Smith *et al.* (1985) and Smith *et al.* (1987). More specialized technical accounts can be found in Naylor and Smith (1982) and Shaw (1985, 1986a, 1986b). Applications of the kinds of techniques described here can be found in Naylor and Smith (1983), Skene (1983), Skene *et al.* (1986), Racine *et al.* (1986) and Shaw (1987).

We shall first describe an iterative quadrature strategy that has proved effective for problems involving up to six parameters. It is well known that univariate integrals of the type

$$\int_{-\infty}^{\infty} e^{-t^2} f(t)\,dt \tag{2}$$

are often well-approximated by Gauss-Hermite quadrature rules of the form

$$\sum_{i=1}^{n} w_i f(t_i), \tag{3}$$

where t_i is the ith zero of the Hermite polynomial $H_n(t)$. In particular, if $f(t)$ is a polynomial of degree at most $2n-1$, then (3) approximates (2) without error. It follows that, if $h(t)$ is a suitably well-behaved function and

$$g(t) = h(t)(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right\}, \tag{4}$$

then

$$\int_{-\infty}^{\infty} g(t)\,dt \approx \sum_{i=1}^{n} m_i g(z_i), \tag{5}$$

where

$$m_i = w_i \exp(t_i^2)\sqrt{2}\sigma, \qquad z_i = \mu + \sqrt{2}\sigma t_i \tag{6}$$

(see Naylor and Smith, 1982). We see, therefore, that, expressed in informal terms, Gauss-Hermite rules are likely to prove very efficient for functions which closely resemble 'polynomial × normal' forms. In fact, this is a rather rich class which, even for *moderate* n (≤ 11, say), *covers many of the likelihood × prior shapes we typically encounter for parameters defined on* $(-\infty,\infty)$. Moreover, the applicability of this approximation is vastly extended by working with suitable transformations of parameters defined on other ranges, such as $(0,\infty)$ or (a,b), using, for example, $\log(t)$ or $\log(t-a)-\log(b-t)$, respectively. Of course, to use (5) we must specify μ and σ in (6). It turns out that, given reasonable starting values (from any convenient source; prior information, maximum likelihood estimates etc), we can successfully iterate on (5), substituting into (6) estimates of the posterior mean and variance obtained using (5) based on previous values of m_i and z_i. Moreover, we note that if the posterior density is well-approximated by the product of a normal and a polynomial of degree at most $2n-3$, then an n-point Gauss-Hermite rule will prove effective for *simultaneously evaluating the normalizing constant and the first and second moments*, using the same (iterated) set of m_i and z_i. In practice, it is efficient to begin with a small grid size ($n = 3$ or $n = 4$) and then to gradually increase the grid size until stable answers are obtained both within and between the last two grid sizes used.

Our discussion so far has been for the one dimensional case. Clearly, however, the need for an efficient strategy is most acute in higher dimensions. The 'obvious' extension of the above ideas is to use a cartesian product rule giving the approximation

$$\int \cdots \int f(t_1,\ldots,t_k)\,dt_1 \ldots dt_k \approx \sum_{i_k} m_{i_k}^{(k)} \cdots \sum_{i_1} m_{i_1}^{(1)} g(z_{i_1}^{(1)},\ldots,z_{i_k}^{(k)}), \tag{7}$$

where the grid points, $z_{i_j}^{(j)}$, and the weights, $m_{i_j}^{(j)}$ are found from (6), substituting the iterated estimates of μ and σ^2 corresponding to the marginal component t_j.

The problem with this 'obvious' strategy is that the product form is only efficient if we are able to make an (at least approximate) assumption of posterior independence among the individual components.

To overcome this problem, we first apply individual parameter transformations of the type discussed above, then we attempt to transform the resulting parameters, via an appropriate linear transformation, to a new, approximately orthogonal, set of parameters. At the first step, this linear transformation derives from an initial guess or estimate of the posterior covariance matrix (for example, based on the observed information matrix from a maximum likelihood analysis). Successive transformations are then based on the estimated covariance matrix from the previous iteration.

We are led to the following general strategy.

1) Reparametrize individual parameters so that the resulting working parameters all take values on the real line.
2) Using initial estimates of the joint posterior mean vector and covariance matrix for the working parameters, transform further to a centred, scaled, more 'orthogonal' set of parameters.
3) Using the derived initial location and scale estimates for these 'orthogonal' parameters, carry out, on suitably dimensioned grids, cartesian product integration of functions of interest.
4) Iterate, successively updating the mean and covariance estimates, until stable results are obtained both within and between grids of specified dimension.

We now describe an iterative importance sampling strategy which has proved effective in higher dimensions. The importance sampling approach to numerical integration is based on the observation that, if f and g are density functions,

$$\int f(x)\,dx = \int [f(x)/g(x)]g(x)\,dx$$
$$= \int [f(x)/g(x)\,dG(x) = E_G[f(X)/g(X)],$$

which suggests the 'statistical' approach of generating a sample from the distribution G and using the average of the values of the ratio f/g as an unbiased estimator of $\int f(x)\,dx$. However, the variance of such an estimator clearly depends critically on the choice of G, it being desirable to choose g to be 'similar' to f.

In the univariate case, if we choose g to be heavier-tailed than f, and if we work with $Y = G(X)$, the required integral is the expected value of $f[G^{-1}(X)]/g[G^{-1}(X)]$ with respect to a uniform distribution on the interval $(0, 1)$. Owing to the periodic nature of the ratio function over this interval, we are likely to get a reasonable approximation to the integral by simply taking some equally spaced set of points on $(0, 1)$, rather than actually generating 'uniformly distributed' random numbers. If f is a function of more than one argument (k, say), an exactly parallel argument suggests that the choice of a suitable g followed by the use of a suitably 'uniform' configuration of points in the k-dimensional unit hypercube will prove an acceptable alternative to the 'costly' procedure of generating 'random' uniformly distributed points in k-dimensions.

However, the effectiveness of all this depends on choosing a suitable G, bearing in mind that we need to have available a flexible set of possible distributional shapes, for which G^{-1} is available explicitly. In the univariate case, such a family defined on $\mathbb{R}$ is provided by considering the random variable

$$X_A = Ah(U) + (1 - A)h(1 - U),$$

where U is uniformly distributed on $(0, 1)$, $h: (0, 1) \to \mathbb{R}$ is a monotone increasing function such that $\lim_{u \to 0} h(u) = -\infty$ and $0 \leq A \leq 1$ is a constant. The choice $A = 0.5$ leads to symmetric distributions; as $A \to 0$ or $A \to 1$ we obtain increasingly skew distributions (to the left or right). The tail-behaviour of the distribution is governed by the choice of the function h. Thus, for example, $h(u) = \log(u)$ leads to a family whose symmetric member is the logistic distribution; $h(u) = -\tan[\frac{1}{2}\pi(1 - u)]$ leads to a family whose symmetric member is the Cauchy distribution. Moreover, the moments of the distribution are polynomials in A (of corresponding order), the median is linear in A, etc, so that sample information about such quantities provides (for any given choice of h) operational guidance on the appropriate choice of A. For details, see Shaw (1986a). To use this family in the multiparameter case, we again employ individual parameter transformations, so that all parameters belong to $\mathbb{R}$, together with 'orthogonalizing' transformations, so that parameters can be treated 'independently'. In the transformed setting, it is natural to consider an iterative importance sampling strategy which attempts to learn about an appropriate choice of G for each parameter.

As we remarked earlier, part of this strategy requires the specification of 'uniform' configurations of points in the k-dimensional unit hypercube. This problem has, in fact, been extensively studied by number theorists and systematic experimentation with various suggested forms of 'quasi-random' sequences has identified effective forms of configuration for importance sampling purposes. For details, see Shaw (1986b). The general strategy is then the following:

1) Reparametrize individual parameters so that the resulting working parameters all take values on the real line.
2) Using initial estimates of the joint posterior mean vector and covariance matrix for the working parameters, transform further to a *centred, scaled, more 'orthogonal' set of parameters*.
3) In terms of these transformed parameters, set $g(\cdot) = \prod_{j=1}^{k} g_j(\cdot)$ for 'suitable' choices of g_j, $j = 1, \ldots, k$.
4) Use the inverse *cdf* transformation to reduce the problem to that of calculating an average over a 'suitable' uniform configuration in the k dimensional hypercube.
5) Use information from this 'sample' to learn about skewness, tailweight etc for each $j = 1, \ldots, k$ and hence choose 'better' g_j, $j = 1, \ldots, k$; as well as revising estimates of the mean vector and covariance matrix.
6) Iterate until the sample variance of replicate estimates of the integral value is sufficiently small.

In reconstructing either a univariate density or the contours of a bivariate density, we begin with a set of density values at some set of parameter values. In the context of the product rule quadrature approach, the parameter values will correspond to grid points selected by a quadrature rule. In the context of importance sampling, the resulting configuration of spot-heights would typically be too irregular for efficient graphical reconstruction and so a mixed strategy is adopted, using a quadrature approach for the parameters of interest and sweeping out the others by importance sampling.

In either case, the approach we adopt for the parameters of interest is to fit splines to the logarithms of the density values. For univariate reconstruction we use 'not-a-knot' cubic splines; for contouring, we use tensor product splines. See also Smith *et al.* (1985) and, for a much more detailed account, Shaw (1985).

The strategies outlined in this paper depend heavily on the availability of interactive computing facilities with graphics capabilities. At the time of writing (and for the foreseeable future), rapid changes are taking place in both technical and economic aspects of the availability of appropriate computing environments. The direction of these changes will clearly influence the form in which Software for Practical Bayesian Statistics will be packaged and marketed, and this applies, in particular, to the software relating to these strategies. For the present, anyone interested in obtaining some form of this software should contact the author.

Acknowledgements

The work described here was largely developed under the auspices of a grant from the UK Science and Engineering Research Council.

References

NAYLOR J. C. and SMITH A. F. M. (1982). Applications of a method for the efficient computation of posterior distributions. *Applied Statistics*, **31**, 214–225.

NAYLOR J. C. and SMITH A. F. M. (1983). A contamination model in clinical chemistry. In *Practical Bayesian Statistics* (eds. A. P. David and A. F. M. Smith). Longman. Harlow.

RACINE A., GRIEVE A. P., FLÜHLER, M. and SMITH A. F. M. (1986). Bayesian methods in practice: experiences in the pharmaceutical industry (with Discussion). *Appl. Statist.*, **35**, 93–150.

SHAW J. E. H. (1985). A strategy for reconstructing multivariate probability densities. *Statistics Group Technical Report 05-85*, Department of Mathematics, University of Nottingham.

SHAW J. E. H. (1986a). A class of univariate distributions for use in Monte Carlo studies. *Statistics Group Technical Report 04-86*, Department of Mathematics, University of Nottingham.

SHAW J. E. H. (1986b). A quasirandom approach to integration in Bayesian Statistics. *Statistics Group Technical Report 05-86*, Department of Mathematics, University of Nottingham.

SHAW J. E. H. (1987). Numerical Bayesian analysis of some flexible regression models. *The Statistician*, **36**, 147–154.

SKENE A. M. (1983). Computing marginal distributions for the dispersion parameters of analysis of variance models. In *Practical Bayesian Statistics* (eds. A. P. David and A. F. M. Smith). Longman. Harlow.

SKENE A. M., SHAW J. E. H. and LEE T. D. (1986). Bayesian modelling and sensitivity analysis. *The Statistician*, **35**, 281–288.

SMITH A. F. M., SKENE A. M., SHAW J. E. H. and DRANSFIELD M. (1985). The implementation of the Bayesian paradigm. *Commun. Statist.*, **A14**, 1079–1102.

SMITH A. F. M., SKENE A. M., SHAW J. E. H. and NAYLOR J. C. (1987). Progress with numerical and graphical methods for practical Bayesian statistics. *The Statistician*, **36**, 75–82.

A BAYESIAN APPROACH TO THE DESIGN AND ANALYSIS OF COMPUTATIONAL EXPERIMENTS

Toby J. Mitchell and Max D. Morris, Oak Ridge National Laboratory

0. Abstract

In a computational experiment, the data are produced by a computer program that models a physical system. The experiment consists of a set of model runs; the design of the experiment specifies the choice of program inputs for each run. This paper centers on the problem of prediction (interpolation), the goal of which is to devise a design/analysis method which will provide predictions of model ouput for input values not run. We adopt a Bayesian approach as the basis for the analysis. Uncertainty about the response function is quantified by choosing a class of probability distributions over the function space. This leads to design procedures based on maximizing the expected reduction in "amount of uncertainty", where the latter can be defined formally in terms of properties of the posterior distribution. Here we use as a design optimality criterion the determinant of the posterior covariance matrix of the responses at the input configurations at which we want to make predictions. This requires maximization of the determinant of the prior covariance matrix of the responses at the design sites. We describe our computer algorithm for constructing optimal designs, and give some examples of designs that it produces.

1. Introduction

1.1 Computer models and computational experiments.

There is widespread and growing use of computer models as tools in scientific research. As surrogates for physical or behavioral systems, computer models can be subjected to experimentation, the goal being to predict how the corresponding real system would behave under certain conditions. This paper is motivated by the goal of getting information from computer models as efficiently as possible.

Here we regard a computer model as a computer program that maps a vector of input variables (parameters) t into a vector of output variables y, where t and y are physically meaningful. We view y as a function $y(t)$ over some domain T in the space of the input variables. This function is deterministic: if the program is run twice (on the same computer) with the same value of t, the same value of y will result.

We consider a *computational experiment* to be a collection of runs of the computer model, made for the purpose of investigating $y(t)$ for $t \in T$. For convenience, we shall consider T to be defined only by the *design variables*, i.e., those variables that are changed during the course of the experiment. In a typical experiment of n runs, the i^{th} computer run is made using inputs $t_i \in T$, $i = 1, 2, \ldots, n$; this collection of input configurations is called the *experimental design*.

There are several important general classes of problems that can be approached through computational experiments, e.g., prediction, sensitivity analysis, uncertainty analysis, optimization, root finding, and integration of output. Perhaps the most fundamental is the problem of prediction of $y(t)$ at sites t that have not been directly observed. The design of experiments for this purpose is the subject of this paper.

We consider a solution to the prediction problem to include a prediction equation $\hat{y}(t)$, formulas for evaluating the uncertainty of prediction, and rules for choosing the design sites. Because of the nature of our approach, which is described below, our method is quite similar to interpolation, in that the prediction of y will be identical to the observed y at values of t for which the model has been run. At other values of t, our prediction will take the form of a probability distribution, the mean of which, expressed as the function $\hat{y}(t)$, can be used as a prediction equation.

We approach the problem from a Bayesian point of view, under which uncertainty about the function y is expressed by means of a probability distribution over all possible response functions. Random functions (stochastic processes, random fields) have been used as models in kriging and other spatial applications for a long time, not generally in an overt Bayesian sense, however. The prediction problem in spatial settings is usually formulated as the problem of making inferences about the realization of a spatial stochastic process $Y(t)$, given the values of that process at a set of "sites" $t_1, \ldots, t_n$. See Ylvisaker (1987) for a discussion of problems of this general type and of the associated design problems. Recently, Shewry and Wynn (1986) and Sacks and Schiller (1987) proposed and used design optimality criteria based on spatial stochastic process models to compute optimal designs for prediction in various settings. Kimeldorf and Wahba(1970) were the first, as far as we know, to use a stochastic process in an explicitly Bayesian sense, for the purpose of predicting a fixed but unknown function. Only recently has there emerged an interest in applying stochastic process models to the design and analysis of computational experiments (Sacks, Schiller, and Welch, 1988).

In this paper, we shall focus on the problem of designing computational experiments for prediction. We present our approach to prediction, given a design, briefly in Section 2. In Section 3, we describe a design criterion and our algorithm for constructing designs that are optimal with respect to it.

2. Prediction

We represent "knowledge" about the unknown function $y(t)$ by a stochastic process $Y(t)$, where

(P1). $Y(t)$ has a normal distribution with mean μ and variance σ^2 (the same for all t), and

(P2). For any pair of sites $t \in T$, $s \in T$, the correlation between $Y(t)$ and $Y(s)$ is a function only of the vector of differences $d = t - s$, i.e.,

$$\rho_{ts} = Corr(Y(t), Y(s)) = R(t-s) = R(d), \qquad (2.1)$$

where $R(d) = R(-d)$ and $R(0) = 1$.

The posterior distribution of Y on any finite set in T, given the set of observed responses $y(D)$ on the set of design sites D, is easily obtained as a conditional multivariate normal distribution.

Let

$$C_D = Corr(Y(D), Y(D))$$

be the $n \times n$ matrix whose elements are the prior correlations between the responses at all pairs of design sites. Let

$$r_D(t) = Corr(Y(t), Y(D))$$

be the n-vector of prior correlations between $Y(t)$ and $Y(D)$.

Then the posterior distribution of $Y(t)$ is normal with mean:

$$\mu_{t|D} = \mu + r_D^T(t) C_D^{-1} (y_D - \mu J) \qquad (2.2)$$

and variance

$$\sigma^2_{t|D} = \sigma^2 [1 - r_D^T(t) C_D^{-1} r_D(t)], \qquad (2.3)$$

where J in (2.2) is an n-vector of 1's, and y_D is the set of observed responses $y(D)$ written as vector. The posterior covariance of $Y(t)$ and $Y(s)$ is

$$\Sigma_{ts|D} = \sigma^2 [\rho_{ts} - r_D^T(t) C_D^{-1} r_D(s)]. \qquad (2.4)$$

All knowledge about $y(t)$ given the data and the prior process is embodied in the posterior process defined by (2.2)-(2.4), which is Gaussian like the prior process, but is no longer stationary. Since we shall use the posterior process for prediction, we shall often refer to it as the "predictive process". The mean of this process (2.2), viewed as a function of t, can be taken as $\hat{y}(t)$; this is an interpolating function, since it passes through the observed y's. The posterior variance (2.3) can be used as a measure of uncertainty of prediction at site t; it is necessarily 0 at the observed sites.

3. Design

3.1 Design Criterion

Suppose we want to design an experiment in n runs for prediction at a finite set of n^* sites $T^* \subset T$, where $n^* > n$. After the experiment is run, knowledge of y at these sites will be embodied in the n^*-dimensional normal distribution of $Y(T^* | D)$ generated by the predictive process there. The mean $\mu_{T^*|D}$ and the covariance matrix $\Sigma_{T^*|D}$ of this distribution can be obtained using (2.2)-(2.4).

We shall adopt as our design criterion the minimization of the determinant of $\Sigma_{T^*|D}$. We refer to this criterion as D-optimality because, like the usual D-optimality criterion in the linear model setting, its goal is to minimize the posterior generalized variance of the unknowns that one is trying to estimate. Shewry and Wynn(1986) have shown that this is equivalent to maximizing the expected gain in information (Lindley, 1956), where information is measured by Shannon's entropy. Shewry and Wynn also showed that this is equivalent to maximizing the determinant of C_D.

Given a correlation function, a D-optimal design can, in principle, be found before any data on y are taken, since the optimality criterion does not depend on y. Except in a few special cases, however, there seem to be few theoretical results available for finding such designs. The designs constructed for this paper were obtained from a computer algorithm adapted from DETMAX (Mitchell, 1974), which was first developed for the purpose of constructing D-optimal designs for linear regression. The optimization method is based on a series of "excursions", which are sequences of designs in which each design differs from its predecessor by the presence or absence of a single site. The first and last designs in an excursion have n sites; the intermediate designs all have fewer sites. (This restriction to designs with n or fewer sites was put in to avoid numerical problems associated with the nearly singular C_D matrices that sometimes arose when the number of sites became large. It ensures that C_D for any design D encountered during the excursion is at least as well conditioned as the starting design.)

The first step of each excursion removes a site from the best current design. At subsequent steps, a site is added, unless the design at that step has already been declared a "failure design", in which case a site is removed. (All designs encountered since the most recent successful excursion are designated as failure designs.) For the purpose of checking a design for equivalence to a failure design, only the determinants of their correlation matrices are compared; thus false equivalence may occasionally be declared. All additions and deletions are made with the goal of maximizing the determinant of the correlation matrix for the resulting design. By this criterion, the best site t to add to an existing design D is the one at which the variance function $\sigma^2_{t|D}$ is greatest. It can also be shown that the largest determinant after deletion of a site in D can be achieved by choosing that site to be the one associated with the greatest element of the diagonal of C_D^{-1}.

The search for the best site to add, i.e., the t at which the variance function (2.3) is maximized, is conducted over a grid in T. Except when T has few dimensions or the grid is very coarse, it is not practical to make the search exhaustive. Instead we have incorporated a multiple search procedure that can best be envisioned by thinking of a set of n hikers trying to climb a hill. Each hiker starts at one of the n current design sites; at each of these, the variance function is zero. The algorithm proceeds by stages, where in each stage, each hiker takes one step in the direction that allows him to increase his altitude the most. We restrict him to consider only the $2k$ neighboring grid points associated with a change in exactly one of the k design variables, and of course we don't let him step outside of T. Under this procedure, the variance function (2.3) is evaluated at (at most) $2nk$ sites in each stage. Sometimes, two hikers will merge, in which case they continue as one. The search ends when all hikers have stopped at (local) maxima; the site that corresponds to the largest of these is taken to be the best site to bring into the design at the current point in the excursion.

The number of excursions made during each search ("try") is determined by restricting the maximum allowed deviation from the nominal number of runs (n), the maximum allowed number of successive excursions that fail to improve $|C_D|$, and the maximum allowed number of "failure designs". (We generally set these restrictions to 4, 10, and 20, respectively.) When one of these constraints causes the search to end, a check for local optimality is made by removing each design site in turn and attempting to replace it by another, using the "hikers" algorithm. If the latter succeeds in finding the global maximum of the variance function in each case, then D is locally optimal in the sense that it cannot be increased by moving a single site. However, the success of the "hikers" algorithm is not guaranteed, and even if it were, the search would not necessarily produce a global optimum.

Table 1 gives an example of a design (on a 6^5 grid in the 5-dimensional unit hypercube) generated by our algorithm for the case $n = 6, k = 5$, under a "product linear" correlation function:

$$Corr(Y(t), Y(s)) = R(d) = \prod_{j=1}^{k} (1 - (1 - \rho_j)|d_j|),$$

where $d_j = t_j - s_j$ and $\rho_j = 0.99, j = 1, \ldots, k$. (When generating designs in the absence of previous data, we usually choose the correlation function to be a product of identical one-dimensional correlation functions.)

Table 1. Allegedly D-optimal design in 5 variables and 6 runs.

Site No.	t_1	t_2	t_3	t_4	t_5
1	0.0	0.0	0.0	0.0	0.0
2	0.6	0.0	1.0	1.0	0.0
3	0.0	0.6	1.0	0.0	1.0
4	1.0	1.0	0.6	0.0	0.0
5	1.0	0.0	0.0	0.6	1.0
6	0.0	1.0	0.0	1.0	0.6

This design exhibits some interesting geometrical structure. Each of the sites in set $A = \{ 2, 3, 4, 5, 6 \}$ is at distance 2.8 from its two nearest neighbors in A and at distance 3.2 from its two most distant neighbors in A, and each site in A is at distance 2.6 from site 1. (Here "distance" is measured along the grid.) Because of the high value of ρ, there is a large region in the middle of T in which there are no design sites; predictions here rely heavily on information from the surrounding design sites. This characteristic is even more pronounced for smoother correlation functions. If we use the cubic correlation:

$$R(d) = \prod_{j=1}^{k} [1 - (a/2)d_j^2 + (b/6)|d_j|^3]$$

with a and b chosen so that, if s and t are at opposite corners of the 5-cube T, $Corr(Y(t), Y(s)) = Corr(Y'(t), Y'(s)) = 0.99^5$, all six sites in the optimal design are on corners of T. In fact, this design turns out to be equivalent to the D-optimal first order regression design in 5 factors and 6 runs (Galil and Kiefer, 1980).

At the other extreme, designs that infiltrate T to a greater extent can be constructed by using correlation functions $R(d)$ that decrease rapidly with $|d|$. For example, consider the correlation function:

$$R(d) = \prod_{j=1}^{k} \rho^{d_j}$$

with $\rho = 0.0001$. The best 16-run design (on a 20×20 grid in the unit square) produced by our algorithm in 10 tries is shown in Figure 1. All

ten tries gave slightly different determinant values, so it is unlikely that this design is truly optimum. There seemed to be little point in undertaking more tries, however, since the computing time per try was about 45 seconds on a Cray X-MP. We did try various grid sizes, to avoid penalizing ourselves by choosing too coarse a grid. We found that 20×20 was sufficient; finer grid sizes require increasingly longer computation time with little apparent benefit.

We favor this kind of design as an initial design in a stagewise approach, in which the correlation function that is used to generate the design sites at each stage may change during the course of the experiment. Methods for selecting the correlation function via cross-validation are discussed in Currin, Mitchell, Morris, and Ylvisaker (1988); some applications to computer models are given there also as examples. and applications are given there also.

4. Acknowledgements

This work has benefitted from our many conversations with Prof. Jerry Sacks of the University of Illinois, Prof. William Welch of the University of Waterloo and the University of Illinois, Prof. Henry Wynn of City University, London, and Prof. Don Ylvisaker of UCLA. This collaboration was started, and continues to be nurtured, by a series of workshops on efficient data collection, funded by a National Science Foundation grant (NSF DMS 86-09819).

5. References

Currin, C., Mitchell, T.J., Morris, M.D., and Ylvisaker, D. (1988), "A Bayesian Approach to the Design and Analysis of Computational Experiments", ORNL report (to appear).

Galil, Z. and Kiefer, J. (1980), "D-Optimum Weighing Designs", Annals of Statistics 8, 1293-1306.

Kimeldorf, G.S. and Wahba, G. (1970), "A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines", Ann. Math. Statist., 41, 495-502.

Lindley, D.V. (1956), "On a Measure of the Information Provided by an Experiment", Ann. Math. Statist. 27, 986-1005.

Mitchell, TJ. (1974), "An Algorithm for the Construction of 'D-Optimal' Experimental Designs", Technometrics, 16, 203-210.

Sacks, J. and Schiller, S. (1987), "Spatial Designs", Fourth Purdue Symposium on Statistical Decision Theory and Related Topics, ed. S.S. Gupta, Academic Press (to appear).

Sacks, J., Schiller, S.B., and Welch, W.J. (1987), "Designs for Computer Experiments", Technical Report #1, Department of Statistics, University of Illinois.

Shewry, M.C. and Wynn, H.P. (1986), "Maximum Entropy Sampling", Technical Report No. 2, The Statistical Laboratory, City University, London.

Ylvisaker, D. (1987), "Prediction and Design", Ann. Statist., 15, 1-19.

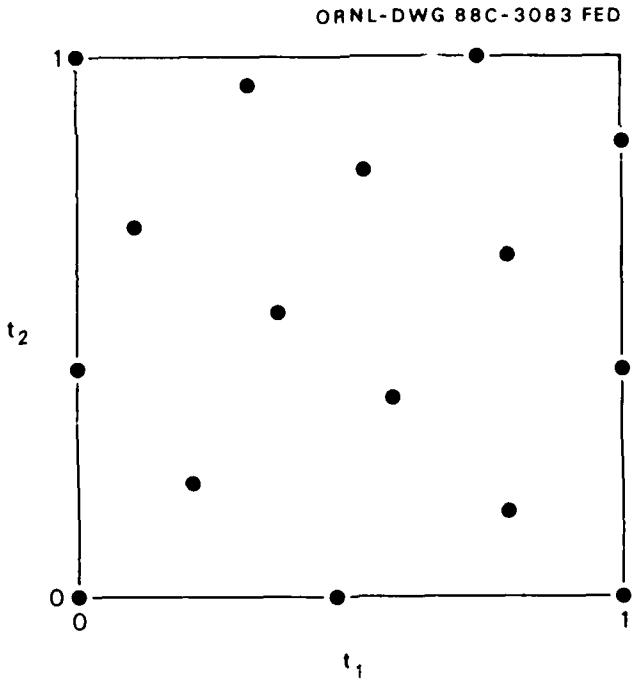


Figure 1. Best 16-run design found on a 20 × 20 grid. using an exponential correlation function with $\rho = 0.0001$.

Additive Principal Components : a Method for Estimating Additive Equations with small Variance

Deborah J. Donnell*
Bellcore.

ABSTRACT

Additive Principal Components are a generalization of linear principal components, where the usual linear function, $a_i X_i$, defining the linear principal component, $\sum_i a_i X_i$, is replaced by a possibly non-linear function, $\phi_i(X_i)$, to form an additive principal component $\sum_i \phi_i(X_i)$. We investigate the analogy to the smallest linear principal component. We present two approaches to estimation — a finite dimensional method, based on a matrix eigen decomposition, and an iterative algorithm, based on a componentwise minimization scheme.

The smallest additive principal component describes nonlinear structure in a high-dimensional space. Consequently it is difficult to interpret the estimated functions in terms that are meaningful for the data analyst. For the additive principal component, the task of interpretation is almost intractable without tools for real time graphical interaction. With these tools, a pleasingly direct method for interpretation of the functions in terms of the original variables is possible.

1 INTRODUCTION

In this paper we investigate the additive analogue to the *smallest* principal component, that is, we estimate additive functions from multivariate data which satisfy as nearly as possible the constraint :

$$\sum_{j=1}^{p} \phi_j(X_j) = 0.$$

Such an additive constraint describes high-dimensional structure in the data. Recall the linear structure implied by a linear constraint, $l(x) = a \cdot x = 0$. If the data nearly satisfy this constraint, they lie close to a linear manifold of co-dimension $p-1$. Analogously, an additive constraint defines an additive manifold of co-dimension 1, and data nearly satisfying this constraint lie near this additive manifold.

Estimation of constraints is an appropriate analysis tool when the search for structure in the data is undirected, that is, no variables are designated *a priori* as predictors of a response of interest. Hence it is a valuable exploratory tool for investigating dependencies in multivariate observational data, where variables are usually interdependent.

Additive principal components were first considered in the context of detecting instability in the additive regression model. The importance of recognizing nonlinear dependencies among the predictor variables when fitting additive regression models is analogous to the importance of detecting collinearity patterns when fitting linear models (Silvey 1969). Suppose we were to fit an additive model $Y \approx \sum_{j=1}^{p} \theta_j(X_j)$ to the data, when there is an exact *concurvity* between the predictors, that is, there are functions of the variables such that $\sum \phi_i(X_i) = 0$. In this situation, the alternative fit :

$$Y \sim \sum_{j=1}^{p} (\theta_j + \phi_j)(X_j),$$

is indistinguishable from the initial one. While exact concurvity is unlikely, even if the data come close to satisfying this constraint, some or all of the estimated θ_j are likely to be unstable. A method which enables us to examine how close the data come to satisfying an additive constraint would thus be a diagnostic check for global stability of the transforms in additive or ACE regression.

Additive principal component analysis is closely related to multiple correspondence analysis (Benzecri 1972, Gifi 1981), and to the nonlinear principal components of De Leeuw (1982a), both of which consider largest principal components of a transformation of the variables. These techniques have been developed and used primarily with psychometric data; their relationship to

*Post-Doctoral Member of Technical Staff, Statistics Research Group, Bellcore, 445 South Street, Morristown, NJ 07960-1910. This work was partially supported by the Department of Energy under Contract DE-FG06-85ER25006

APC analysis is discussed in Section 4.

Following the formal definition of the APC, we give a brief derivation of its characterization as an eigenfunction of a compact operator in Section 3. In Section 5 we discuss methods of estimation.

The final sections are of a more practical nature, concerned with using APC analysis as an applied method : Section 6 discusses interpretation techniques which we use to interpret the smallest APC of a data set in Section 7.

2 THE POPULATION ADDITIVE PRINCIPAL COMPONENT

2.1 Motivation

Strong additive dependence in a set of variables exists if the data can be transformed so that $X_1, X_2, , \ldots, X_p$ come close to satisfying an additive constraint $\sum \phi_i(X_i) = 0$. Our objective is to characterize the set of unknown transformations $\phi_1, \phi_2, \ldots, \phi_p$.

When the transformations are restricted to be linear, we simply have the classical problem of the analysis of collinearity. The simple rationale that a linear combination of the variables with variance near zero implies the variables are nearly collinear, leads to the criterion :

$$\min_{\mathbf{a} \in \mathcal{R}^p} \mathbf{var}\left(\sum a_i X_i\right) \text{ subject to } \sum a_i^2 = 1.$$

The variables are usually standardized, somewhat arbitrarily, to have $\mathbf{E}(X_i) = 0$ and $\mathbf{var}(X_i) = 1$.

The minimum occurs for $\mathbf{a}$ an eigenvector for the smallest eigenvalue of $\mathbf{cov}(\mathbf{X}) = \mathbf{\Sigma}$, and the random variable $\sum a_i X_i$ is known as a smallest principal component of $\mathbf{X}$. A geometric characterization of the smallest principal component comes from observing that the linear function, $l_1(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$, of the minimizing vector $\mathbf{a}$, defines a linear manifold of co-dimension 1 in p-space through $l_1(\mathbf{x}) = 0$. This linear manifold minimizes the expected squared distance from the observations to any linear manifold of co-dimension 1.

In short, there are three characterizations of the vector $\mathbf{a}$ defining the principal component :

- $\sum a_i X_i$ has minimal variance among all linear combinations of the variables with $\sum a_i^2 = 1$.
- $\mathbf{a} \cdot \mathbf{x} = 0$ defines the manifold of co-dimension 1 minimizing expected squared distance to the data.
- $\mathbf{a}$ is an eigenvector for the smallest eigenvalue of $\mathbf{\Sigma}$.

A natural approach for a generalization of principal components to additive functions is to extend one of these definitions of the smallest linear principal component of $\mathbf{X}$. The minimum variance characterization suggests defining $\Phi = (\phi_1, \ldots, \phi_p)$ as the vector of transformations of the variables minimizing $\mathbf{var} \sum \phi_i(X_i)$ subject to some normalizing constraint. Alternatively, a geometric characterization would determine the additive manifold described by the constraint $\sum \phi_i(X_i) = 0$, which minimizes the expected squared distance from the observations to any additive manifold of co-dimension 1. Unlike the linear case, the additive functions determined by the above two definitions will not be the same.

In this paper we use the minimum variance definition, which has two useful characteristics not shared by the geometric approach. First, the minimum variance criterion leads to a characterization of the additive principal component as an eigenfunction, from which we gain a wealth of theoretical insight into the behavior and properties of our estimator. Second, finite sample estimates are easy to compute, since the criterion involves estimation of variance rather than estimation of the euclidean distance between a manifold and the data.

2.2 Definition of the Smallest Additive Principal Component

For simplicity, we will assume each additive function of an APC to be centered, and require the variance to be finite. Formally, the APC-function of the i^{th} variable, $\phi_i(X_i) \in H(X_i)$, where :

$$\begin{aligned} H(X_i) &\subset \{\phi_i : \mathbf{E}\,\phi_i(X_i) = 0,\ \mathbf{var}\,\phi_i < \infty\} \\ &= L_2(X_i). \end{aligned}$$

The vector of APC-functions, $\Phi(\mathbf{X}) = (\phi_1(X_1), \ldots, \phi_p(X_p))$, belongs to the product space defined by the component spaces $H(X_i)$:

$$\begin{aligned} \Phi(\mathbf{X}) &\in H(\mathbf{X}) \\ H(\mathbf{X}) &\stackrel{\text{def}}{=} H(X_1) \times H(X_2) \times \cdots \times H(X_p) \\ &\subset L_2(\mathbf{X}). \end{aligned}$$

Definition 2.1 *The smallest additive principal component of* $\mathbf{X} = (X_1, \ldots, X_p)$ *in* $H(\mathbf{X})$ *is the random variable* $\tilde{\phi}(\mathbf{X}) = \sum_{i=1}^p \phi_i(X_i)$, $\phi_i(X_i) \in H(X_i)$, *minimizing* $\mathbf{var} \sum_{i=1}^p \phi_i(X_i)$ *subject to* $\sum_{i=1}^p \mathbf{var}\,\phi_i(X_i) = 1$.

Note that the constraint, $\sum \mathbf{var}\, \phi_i = 1$, is a natural analogue to the linear constraint, $\sum a_i^2 = 1$, under the usual assumption that all variables have equal variance. For if $\phi_i(X_i) = a_i X_i$, $\sum \mathbf{var}\, \phi_i(X_i) = \sum \mathbf{var}\, a_i X_i = \sum a_i^2 \mathbf{var}\, X_i = \sum a_i^2 = 1$. Restriction of $H(\mathbf{X})$ to linear functions reduces to a definition of linear principal components for the correlation case.

3 THE EIGEN CHARACTERIZATION

3.1 Introduction

In the preceeding section the definition of the APC was presented as a natural extension of the linear principal component to additive functions. It is not unexpected, then, that the eigen characterization of the additive principal component can be derived by considering an extension of the eigen characterization of the linear principal component.

Linear algebra gives the well known equivalence between the statements :

$$\text{minimize } \mathbf{var}\,(\textstyle\sum_i a_i X_i) \text{ subject to } \sum_i a_i^2 = 1$$
$$\text{and}$$
$$\text{minimize } \langle \mathbf{a}, \Sigma \mathbf{a} \rangle \text{ subject to } \|\mathbf{a}\|^2 = 1$$

where $\langle \cdot, \cdot \rangle$ is the usual euclidean inner product in $\mathcal{R}^p$. From the latter statement, it follows that the vector of coefficients $\mathbf{a}$, since it minimizes the bounded, symmetric quadratic form $Q(\mathbf{a}) = \langle \mathbf{a}, \Sigma \mathbf{a} \rangle$, is an eigenvector of Σ. Thus, the linear principal component solution can be solved using standard linear algebraic techniques.

Analogously, the smallest additive principal component can be characterized by either of the following criteria :

$$\begin{array}{lll} \Phi & \text{minimizes} & \mathbf{var}\,(\sum_i \phi_i(X_i)) \\ & \text{subject to} & \sum_i \mathbf{var}\, \phi_i(X_i) = 1 \\ & & \text{and} \\ \Phi & \text{minimizes} & \langle \Phi, \mathbf{P}\Phi \rangle_H \\ & \text{subject to} & \|\Phi\|^2 = 1 \end{array}$$

The inner product, $\langle \cdot, \cdot \rangle$, in the above, is the natural inner product on the product space of the vector of APC-functions, $H(\mathbf{X})$:

$$\begin{aligned} \langle \Phi, \Phi' \rangle_H &= \textstyle\sum_i \langle \phi_i, \phi_i' \rangle \\ &= \textstyle\sum_i \mathbf{cov}\,(\phi_i, \phi_i'), \end{aligned}$$

The corresponding norm is :

$$\|\Phi\|_H^2 = \sum_i \|\phi_i\|^2 = \sum_i \mathbf{var}\, \phi_i.$$

$\mathbf{P}$ is the bounded, symmetric, linear operator of the following lemma.

Lemma 3.1 *The operator* $\mathbf{P} : H(\mathbf{X}) \mapsto H(\mathbf{X})$ *defined by the relationship*

$$\mathbf{var} \sum \phi_i = \langle \Phi, \mathbf{P}\Phi \rangle_H$$

is the mapping :

$$[\mathbf{P}\Phi]_i = \mathbf{E}\left[\sum_j \phi_j \mid X_i\right]$$

$\mathbf{P}$ *is symmetric, non-negative definite and bounded above by* p .

Proof

$$\begin{aligned} \langle \Phi, \mathbf{P}\Phi \rangle_H &= \textstyle\sum_i \langle \phi_i, \left(\mathbf{E}\left[\sum_j \phi_j \mid X_i\right]\right) \rangle \\ &= \textstyle\sum_i \mathbf{cov}\left[\phi_i \left(\mathbf{E}\left[\sum_j \phi_j \mid X_i\right]\right)\right] \\ &= \textstyle\sum_i \mathbf{cov}\left[\phi_i \left(\sum_j \phi_j\right)\right] \\ &= \mathbf{cov}\left[\left(\sum_i \phi_i\right)\left(\sum_j \phi_j\right)\right] \\ &= \mathbf{var}\left(\textstyle\sum_i \phi_i\right). \end{aligned}$$

$\mathbf{P}$ is bounded by p :

$$\begin{aligned} \|\mathbf{P}\Phi\|_H^2 &\stackrel{\text{def}}{=} \textstyle\sum_i \|P_i \sum_j \phi_j\|^2 \\ &\leq \textstyle\sum_i \|\sum_j \phi_j\|^2 \\ &= p\,\|\textstyle\sum_j \phi_j\|^2 \\ &\leq p\left(\textstyle\sum_j \|\phi_j\|\right)^2. \end{aligned}$$

The maximum of $\sum_j \|\phi_j\|$ under the constraint $\sum \|\phi_j\|^2 = 1$ is attained at $\|\phi_j\| = p^{-\frac{1}{2}}$. Hence,

$$\begin{aligned} \|\mathbf{P}\Phi\|_H^2 &\leq p\left(\textstyle\sum_j \|\phi_j\|\right)^2 \\ &\leq p^2. \end{aligned}$$

The inequality is sharp, with equality occurring when $X_i = X_j$, $\phi_i = \phi_j \quad \forall\, i, j$.
Symmetry and non-negativity of $\mathbf{P}$ follow from the properties of $\mathbf{var}\,(\cdot)$. ■

The eigen characterization of the APC now follows almost trivially.

Theorem 3.2 *The smallest eigenfunction of the operator* $\mathbf{P}$, *if it exists, is a vector of APC-functions for the smallest additive principal component of* $\mathbf{X}$.

Proof

Since $\langle \Phi, \mathbf{P}\Phi \rangle_H = \mathbf{var} \sum_i \phi_i$ by Lemma 3.1, and $\sum \mathbf{var}\, \phi_i = \|\Phi\|_H^2$ by definition, a function vector $\Phi \in H(\mathbf{X})$ minimizes $\langle \Phi, \mathbf{P}\Phi \rangle_H$ subject to $\|\Phi\|_H^2$ iff the set of transformations $\{\phi_1, \phi_2, \ldots, \phi_p\}$ minimizes $\mathbf{var} \sum \phi_i(X_i)$ under the constraint $\sum \mathbf{var}\, \phi_i(X_i) = 1$.

From the theory of symmetric operators, Jorgens (1970), Th 6.7 p.125, it is well known that $\Phi \in H(\mathbf{X})$ minimizing $\langle \Phi, \mathbf{P}\Phi \rangle_H$ subject to $\|\Phi\|_H = 1$ is an eigenfunction for the smallest eigenvalue of $\mathbf{P}$ (where it exists) . ■

An immediate corollary to Theorem 3.2 is :

Corollary 3.3 *Suppose $\Phi = (\phi_1, \phi_2, \ldots, \phi_p)$ is a smallest eigenfunction of $\mathbf{P}$ belonging to the smallest eigenvalue λ, with $\|\Phi\| = 1$. Then :*

1. *A smallest APC of $\mathbf{X}$ is $\tilde{\phi} = \sum_i \phi_i$,*
2. *The variance of this smallest APC is λ.*

Proof The first is immediate; for the second,

$$\mathbf{var} \sum \phi_i = \langle \Phi, \mathbf{P}\Phi \rangle_H = \langle \Phi, \lambda\Phi \rangle_H = \lambda \blacksquare$$

A further consequence of the eigenfunction property of the smallest principal component, is the following characterization as a solution of the *APC stationary equations.*

Corollary 3.4 *A smallest additive principal component with variance λ satisfies the stationary equation :*

$$\mathbf{P}\Phi = \lambda\Phi$$

Conversely, any Φ satisfying the stationarity conditions for minimal $\lambda \leq 1$ is a smallest APC of $\mathbf{X}$.

The stationary equation implies a strong set of identities for every APC-function : for each i, the conditional expectation with respect to X_i of the APC is a multiple of the i^{th} APC-function, that is,

$$\mathbf{E}\left(\sum_j \phi_j \mid X_i\right) = \lambda_{min}\phi_i \ \forall i.$$

Moreover, the multiple factor, λ, is constant for all of the APC-functions.

Notice that if the smallest eigenvalue $\lambda_{min} \approx 0$, the conditional expectations of the smallest APC with respect to all variables are almost zero. In this sense we recall our initial motivation: to find functions that come close to satisfying the constraint $\sum_i \phi_i = 0$.

3.2 Infinite Dimensional Function Spaces

We now address the issue of existence of the smallest eigenspace. If $H(\mathbf{X})$ is finite dimensional, the spectrum of the operator $\mathbf{P}$ is discrete, and the smallest eigenspace exists and is distinct. However, for infinite dimensional $H(\mathbf{X})$, although the spectrum of $\mathbf{P}$ is bounded, the existence of the smallest eigenspace is complicated by the possibility of $\mathbf{P}$ having a non-trivial continuous spectrum or spectral values that are not eigenvalues. We can rule out these undesirable possibilities by adopting suitable compactness assumptions, following Breiman and Friedman (1985).

Assumption : *The restricted operators $P_{i/k} : H(X_k) \mapsto H(X_i)$, defined by $P_{i/k}(h(X_k)) = \mathbf{E}(h(X_k) \mid X_i)$ are compact for $k \neq i$, $i = 1, \ldots, p$.*

This assumption is only required for infinite dimensional $H(X_i)$. A sufficient condition for compactness to hold is given in Breiman and Friedman (1985). It is straightforward to show that the assumption of compactness implies the spectrum of $\mathbf{P}$ is essentially discrete, since its continuous spectrum consists of at most one point, namely one. A smallest eigenvalue of one corresponds to the null situation of mutual independence of all variables.

4 RELATED LITERATURE

The idea of using a larger class of functions in principal component analysis is not new: a simple polynomial extension, for instance, appears in the statistical literature in Gnanadesikan (1977). By far the most comprehensive treatment of extensions of principal components analysis, however, are the optimal scaling techniques developed by psychometricians. Multiple correspondence analysis (Benzecri 1972,Lebart et al. 1984) and non-linear principal components analysis (De Leeuw 1982a, Gifi 1981) are techniques, used almost exclusively with categorical data, for determining optimal scalings of the categories — which is equivalent to estimating step functions of the variables — with low dimensional structure.

In this paper we focus on the smallest APC, corresponding to the smallest eigenvalue. In psychometrics, the intended application is an extension of the use of the largest linear principal components for dimension reduction. The largest APCs are clearly interesting in their own right, however their interpretation and potential applications are very different to those of the smallest APC. Nonetheless, we acknowledge that these methods from psychometry use essentially the same notions as additive principal components.

The optimal scalings of multiple correspondence analysis are equivalent to the largest APCs defined over the finite dimensional function space

spanned by normalised indicator functions of the variable categories. Multiple correspondence analysis examines the largest eigenfunctions of the corresponding finite dimensional operator.

The non-linear principal components or PRINCALS (Principal Components by Alternating Least Squares) analysis allows only one set of transformations of the variables, rather than the multiple transformations of multiple correspondence analysis. The transformations are defined to be optimal for some fixed dimensional representation, d, hence for $d = 1$, they are equivalent to multiple correspondence analysis, but for $d \neq 1$, PRINCALS gives a different solution. De Leeuw (1982b) has extended PRINCALS to continuous variables. The functions are estimated using a finite dimensional B-spline basis, hence the problem can be recast as a finite dimensional eigenproblem, solvable by linear techniques.

5 ESTIMATION OF THE ADDITIVE PRINCIPAL COMPONENT

5.1 Introduction

The smallest additive principal component corresponds to the smallest eigenvalue of the symmetric non-negative definite operator $\mathbf{P}$. Thus, for estimation of the additive principal component we turn to known methods for calculating eigenfunctions.

If all the function spaces, $H(X_i)$, of the APC-transforms are finite dimensional, estimation can be simplified to finding the smallest eigenvector of a finite dimensional matrix. We also present an iterative algorithm based on the power method of estimating eigenfunctions, an approach which is valid in the population for general $H(\mathbf{X})$.

We shall not discuss the stability of these estimation methods in depth, but it is important to bear in mind that estimating the smallest eigenfunction is an intrinsically unstable problem when the second smallest eigenvalue is close to the smallest eigenvalue. Any estimation procedure will have difficulty finding a unique, stable estimate of the eigenfunction in this case.

5.2 The direct solution for finite dimensional APC

Assume the function space for the i^{th} variable, $H(X_i)$, is finite dimensional. Then for some finite set of orthogonal basis functions, $\{f_{ik}\}$,

$$H(X_i) \equiv span\{f_{ik}(X_i) : \mathbf{E}\, f_{ik}(X_i) = 0, \mathbf{E}\,(f_{ik}(X_i) f_{ik'}(X_i)) = 0, k = 1, \ldots d_i\}$$

Since $\phi \in H(X_i) \Leftrightarrow \phi_i = \sum_{k=1}^{d_i} a_{ik} f_{ik}(X_i)$, the APC criterion can be written :

$$\begin{aligned} \mathbf{var}\,(\textstyle\sum_i \phi_i) &= \mathbf{var}\,(\textstyle\sum_i \sum_k^{d_i} a_{ik} f_{ik}(X_i)) \\ &= \mathbf{var}\,(\textstyle\sum_i F_i(X_i)\mathbf{a}_i) \\ &= \mathbf{a}^t \mathbf{var}\,(F(\mathbf{X}))\mathbf{a} \\ \text{where } F_i(X_i) &= (f_{i1}(X_i) \ldots f_{id_i}(X_i)), \\ \mathbf{a}_i^t &= (a_{i1}, \ldots, a_{id_i}), \\ F(\mathbf{X}) &= (F_1(X_p), \ldots, F_p(X_p)) \\ \mathbf{a}^t &= (\mathbf{a}_1^t, \ldots, \mathbf{a}_p^t). \end{aligned}$$

The normalising constraint is simply :

$$\begin{aligned} \textstyle\sum \mathbf{var}\,\phi_i &= \textstyle\sum_i \sum_{k=1}^{d_i} \mathbf{var}\, a_{ik} f_{ik} \\ &= \mathbf{a}^t \mathbf{a} = 1. \end{aligned}$$

Estimating the smallest APC simplifies to calculating the smallest linear principal component of the basis vectors, $F(\mathbf{X})$. The smallest APC is the smallest linear principal component of $F(\mathbf{X})$: for the eigenvector $\mathbf{a}$, the APC is $\sum_i \phi_i(X_i) = F(\mathbf{X})\mathbf{a}$; the i^{th} APC-function, $\phi_i(X_i) = F_i(X_i)\mathbf{a}_i$.

Finite sample estimation is straightforward: express each basis vector as a functional of its distribution function, $\mathcal{F}_i$, $f_{ik}(X_i) = \theta_{ik}(\mathcal{F}_i)$. Replacing $\mathcal{F}_i$ with the empirical distribution function, $\mathcal{F}_i^n$, yields finite sample estimates : $\hat{f}_{ik}(x_i) = \theta_{ik}(\mathcal{F}_i^n)$. An APC estimate can be obtained from the eigen decomposition of the correlation matrix of $\hat{F}(\mathbf{x}) = (\hat{f}_{11}(x_1), \hat{f}_{12}(x_2) \ldots \hat{f}_{pd_p}(x_p))$.

5.3 The iterative method

Iterative calculation of the smallest eigenfunction uses a componentwise minimization scheme, where each function is estimated in turn, using the function estimates of the previous iteration. The iterative approach is important both because it enables estimation for a class of functions that are only constrained to be "smooth", and because it provides an alternative to the expense of an eigen decomposition when the dimension of $H(\mathbf{X})$ is large.

5.3.1 A power algorithm

It is easily shown for a symmetric, non-negative operator, that for some initial $\Phi^{(0)}$, the sequence :

$$\frac{\mathbf{P}^k \Phi^{(0)}}{||\mathbf{P}^k \Phi^{(0)}||} \qquad k = 1, 2, \ldots$$

converges a.s. to the eigenfunction of $\mathbf{P}$ belonging to the maximal eigenvalue. This can easily be adapted to find the eigenfunction belonging

to the smallest eigenvalue, since there is a simple linear relationship between the eigenvalues and eigenfunctions of $\mathbf{P}$ and $p\mathbf{I} - \mathbf{P}$.

The eigenvalues of $\mathbf{P}$ are non-negative and bounded above by p. For Θ an eigenfunction of $\mathbf{P}$ with eigenvalue λ,

$$(p\mathbf{I} - \mathbf{P})\Theta = p\Theta - \mathbf{P}\Theta = (p - \lambda)\Theta.$$

It follows that $\mathbf{P}$ and $p\mathbf{I} - \mathbf{P}$ have common eigenfunctions, however the order of eigenvalues for the shifted operator, $p\mathbf{I} - \mathbf{P}$, is reversed. Thus, the sequence :

$$\frac{(p\mathbf{I} - \mathbf{P})^k \Phi^{(0)}}{\|(p\mathbf{I} - \mathbf{P})^k \Phi^{(0)}\|}$$

converges to the smallest eigenfunction of $\mathbf{P}$. The value p can be replaced by the largest eigenvalue of the operator $\mathbf{P}$, in any specific problem, which will improve the rate of convergence dramatically.

The iteration scheme employed is an alternating conditional expectation algorithm, in the same vein as the algorithm used to estimate ACE (Alternating Conditional Expectation) regression. Algorithmically, the sequence is generated as follows :

Algorithm

Choose initial transformations $\phi_1^{[0]}, \phi_2^{[0]}, \ldots, \phi_p^{[0]}$

Repeat 1 and 2 for $N = 1, 2, \ldots$ [*Outer Loop*]

(1)**Do** for $i = 1, \ldots, p$[*Inner Loop*]

$$\phi_i \leftarrow p\phi_i^{[N-1]} - \mathbf{E}\left(\textstyle\sum_{j=1}^{p} \phi_j^{[N-1]} \mid X_i\right).$$

(2)**Standardize**

$$\begin{aligned}(\phi_1^{[N]}, \phi_2^{[N]}, \ldots, \phi_p^{[N]}) &\leftarrow (c\phi_1, c\phi_2, \ldots, c\phi_p) \\ \text{where } c &= \left(\textstyle\sum_i \mathbf{var}\, \phi_i\right)^{-\frac{1}{2}}\end{aligned}$$

Until var $\sum \phi_i^{[N]}$ converges.

Note that while in the ACE regression algorithm, each ϕ_i is updated to its new transformation as the inner loop proceeds, we obtain the new p-tuple *using only the previous p-tuple throughout the entire inner loop.*

The iterative algorithm, as a version of the power algorithm, shares its shortcomings as a method of estimation: it is prone to difficulties associated with finding local, rather than global stationary points, hence it can be sensitive to starting values; convergence will be slow when neighbouring eigenvalues are close.

Obtaining finite sample estimates using the iterative method essentially entails choosing a method for estimating the conditional expectation term of the inner loop. If $H(\mathbf{X})$ is finite dimensional, this is easily done. If $H(\mathbf{X})$ has infinite dimension, approximate solutions are computed using smoothing techniques.

5.3.2 Finite dimensional

For the finite dimensional $H(X_i)$, each conditional expectation operator has the decomposition :

$$\mathbf{E}\,(\phi \mid X_i) = \sum_k \langle \phi, f_{ik} \rangle f_{ik},$$

so each inner loop step is simply a linear least squares regression of $\tilde{\phi} = \sum_i \phi_i$ on $f_{i1} \ldots f_{id_i}$:

$$\mathbf{E}\,(\tilde{\phi} \mid X_i) \quad = \quad \textstyle\sum_k \langle \tilde{\phi}, f_{ik} \rangle f_{ik}.$$

Noting that $f_{ik} \perp f_{ik'}$, it is easily shown the inner product, $\langle \tilde{\phi}, f_{ik} \rangle$ are the coefficients of the linear least squares regression of $\sum_i \phi_i$ on $f_{i1} \ldots f_{id_i}$. Finite sample estimates are obtained in the obvious way, the inner loop step is simply :

$$\mathbf{a}_i^{(\text{new})} \leftarrow p\mathbf{a}_i^{(\text{old})} - \hat{F}_i^{\,t}\left(\sum_{j=1}^{p} \hat{F}_j \mathbf{a}_j^{(\text{old})}\right),$$

and $\phi_i^{(\text{new})}(x_i) = \mathbf{a}_i^{(\text{new})} \hat{F}_i$

5.3.3 Infinite dimensional

A powerful and practical alternative to finite dimensional estimation techniques is estimation of conditional expectations using scatterplot smoothers.

Let S_i denote a smoother with respect to X_i. The inner loop step is implemented as :

$$\phi_i \leftarrow (\tau - 1)\phi_i^{[N-1]} - S_i\left(\sum_{j \neq i} \phi_j^{[N-1]}\right)$$

Since S_i is typically not a projection operator, it is important that $\phi_i^{[N-1]}$ is excluded from the smoothed term. The value τ is an estimate of the largest eigenvalue of $\mathbf{P}$.

The advantages of using a smoother for estimation in terms of flexibility, interpretability and cost are obvious. The disadvantage is that most smoothers are non-linear, hence mathematical analysis of the estimation procedure is usually

not feasible. Our experience, however, matches that of Breiman and Friedman (1985) with the ACE algorithm: with good starting guesses the iterative procedure generally converges to acceptable estimates of the minimizing functions.

We have presented both a direct and iterative method for computing APC estimates, which are equivalent when the function spaces are assumed known and finite. In practice, the iterative algorithm implemented with a scatterplot smoother may be a preferable method, particularly for exploratory analysis, since smoothing techniques place far fewer restrictions on the function space. Unfortunately, justification for this procedure is heuristic for the most part, as smoothers are not usually projection operators. Nevertheless, in the ensuing examples, we use the iterative algorithm with a scatterplot smoother for estimation of the conditional expectation. 1

6 INTERPRETATION OF ESTIMATES

Using the smallest APC as an applied technique for multivariate analysis of a dataset, requires careful consideration of the properties and interpretation of the estimators characterizing the APC: the eigenvalue; the APC; the APC-functions. However, unlike a linear analysis, examining these estimates alone is not sufficient to infer the dependencies between the variables. The APC determines a dependency linear in the *transformed* variables, so unless the transforms themselves are linear, translating this dependence to the original space is far from easy. We suggest a graphical technique using simultaneous highlighting of plots to aid in understanding the concurvity.

6.1 The estimates

1. Eigenvalue : $\mathbf{var}\ (\sum_i \phi_i)$

   The eigenvalue measures the strength of concurvity, and by definition is bounded between 0 and 1 : 0 corresponds to exact concurvity, 1 to mutual independence of transformed variables. The size and spacing of different eigenvalues can warn about potential difficulties with stability and uniqueness : since $\bar{\lambda} = 1$, instability becomes more likely as λ_i approaches 1.

2. APC : $\sum_i \phi_i$

   The smallest APC, by definition, has minimal variance, hence interpretation of the APC vector is akin to a residual analysis. Ideally, it will be distributed symmetrically about zero; departures from symmetry, such as outliers or grouping in the APC, indicate cases which are unusual with respect to the concurvity relation.

3. APC-function weights: $\mathrm{sd}\ (\phi_i(X_i))$

   The relative scale of the APC-transforms, as measured by the APC-function weights, indicates the relative importance of the variable in the APC : a zero weight indicates no contribution, a large weight, a large influence.

4. APC-functions : $\phi_i(X_i)$

   Plotting $\phi_i(X_i)$ versus X_i reveals the shape of the transform, which can indicate the sensitivity of the values of X_i in the dependence : a step function indicates sensitivity only between corresponding levels of the variable, an asymptote defines a region of relative insensitivity.

6.2 *Interpretation using graphics*

Suppose for a data set, $\mathbf{x}$, we have estimated the smallest APC, and its eigenvalue is small, implying near concurvity between the variables. The transformed data, $\Phi(\mathbf{x})$ have the strongest linear dependence achievable —how can we interpret what this linear dependence of transformed variables implies for the relationship between the original variables ? Simultaneous highlighting of scatterplots of the data facilitates the interpretation of the concurvity.

Simultaneous highlighting requires a graphics capability that is most naturally suited to a high resolution graphics terminal equipped with a flexible pointer device, such as a mouse; however it can also be effective with static plots. For a set of plots displaying different variables of the same data set, we want to select any group of cases in any plot and have the selected cases highlighted in all the remaining plots; thus a subset of cases are highlighted simultaneously in all plots. Selection and highlighting are usually indicated by a change in color, size or symbol of the selected cases.

The use of simultaneous highlighting for interpreting an APC is best illustrated by a simple example in three variables.

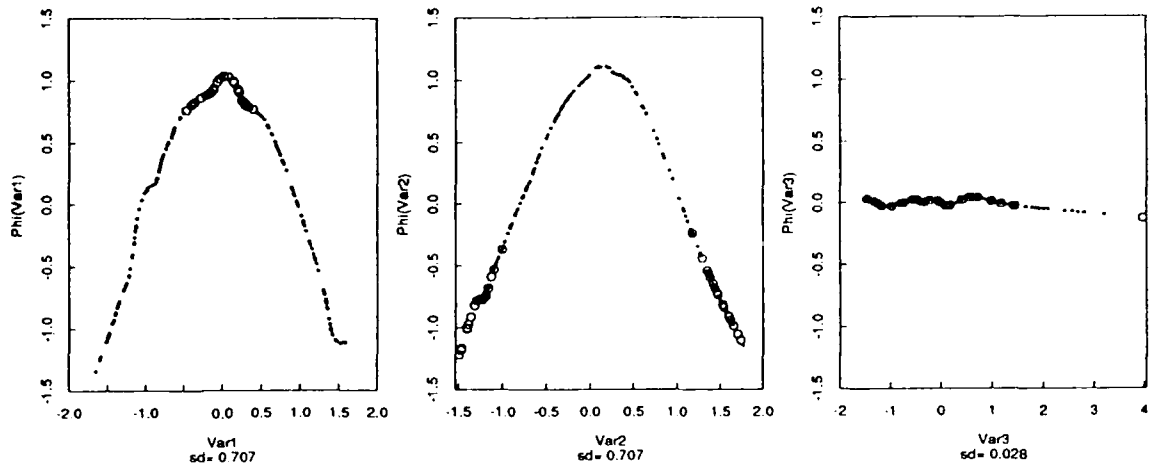


Figure 1: The APC-function plots of the smallest APC. $\mathbf{var}\left(\sum_i \phi_i\right) = 0.084$. Highlighting of large values of $\phi_1(X_1)$ indicates a strong relationship between X_1 and X_2

6.3 An example: Interpretation for a three variable APC

The smallest APC of x_1, x_2, x_3, estimated using the iterative algorithm with the supersmoother, has variance 0.084 — hence the data almost lie on a surface in 3-space. The variable weights are : $\mathbf{sd}\ \phi_1 = 0.71$, $\mathbf{sd}\ \phi_2 = 0.71$, $\mathbf{sd}\ \phi_3 = 0.03$, so we can conclude the third variable is not important in determining the relationship between the variables. The APC-functions are plotted in Figure 1.

For the moment, consider $\phi(\mathbf{x})$ and $\mathbf{x}$ to be distinct data sets : the former has a strong linear structure which we want to use to explore the structure of its untransformed version. Display the two data sets together in the 3 scatterplots : ϕ_1 vs x_1, ϕ_2 vs x_2, ϕ_3 vs x_3; so $\phi(\mathbf{x})$ appears in the horizontal marginal projection, $\mathbf{x}$ in the vertical marginal projection. The small variance of the APC implies ϕ_1, ϕ_2 and ϕ_3 are almost linearly dependent : $\phi_1 + \phi_2 + \phi_3 \approx 0$. As $\mathbf{sd}(\phi_3)$ is small, the values of ϕ_3 are always close to zero, hence low values of ϕ_1 will constrain values of ϕ_2 to be high. Selecting low values of ϕ_1, as in Figure 1, and simultaneously highlighting the selected cases in the other plots illustrates this constraint.

Now, highlighting enables us to use the linear dependence of the transformed variables to reveal the dependence between the original variables. In Figure 1, low values of ϕ_1 occur when x_1 is extreme (either high or low); high values of ϕ_2 when x_2 is central; in the plot of x_3, the selected points are evenly spread along the horizontal axis, confirming the observation that this variable does not determine the concurvity. The constraint $\sum_i \phi_i = 0$, for low ϕ_1, can be interpreted in the variables $\mathbf{x}$: cases with values of x_2 near zero will have either high or low values of x_1. Continuing with selection of cases by conditioning on values over the entire range of ϕ_1, we can understand the configuration of the variables in the original scaling. In this case, the relationship between x_1 and x_2 is easily understood to be circular, hence the variables lie on a cylinder oriented lengthwise along the x_3 axis, Figure 2. In general, conditioning on the values of each transform in turn, much more complex relationships can be explored using this technique.

7 Boston Housing Data

The variables for this example are the variables selected by Breiman and Friedman (1985) as explanatory variables for median housing values in Boston ·

Noxsq Nitrogen Oxide concentration in pphm squared.

Tax Full Property Tax rate

Ptratio Parent Teacher ratio of the town school district

Lstat Proportion of population that is of lower status

Roomsq Average number of rooms squared

The smallest APC of the five variables is estimated, and shown in Figure 3. The variance of the APC is 0.035, hence there is strong evidence that dependencies exist. The estimated

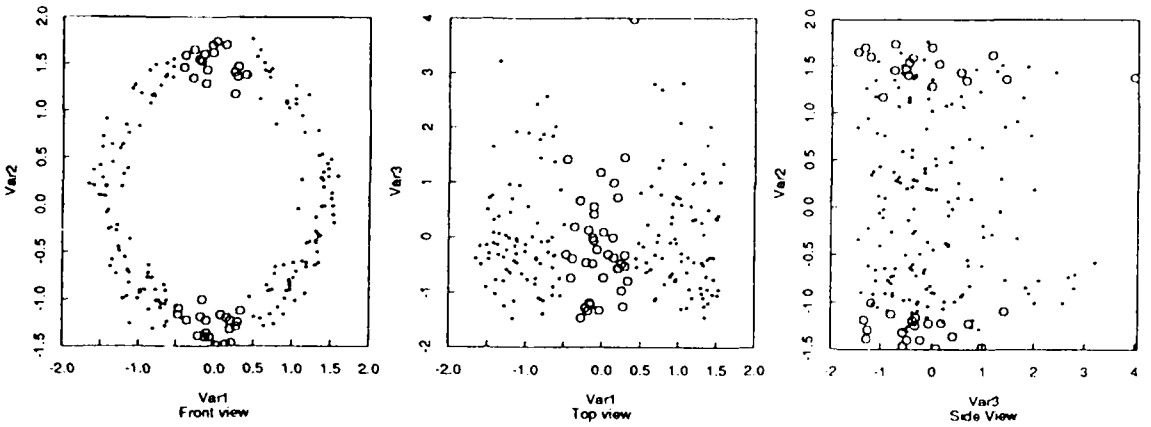


Figure 2: The configuration of the dataset in Example 6.1. The variables lie on the surface of a cylinder

transforms indicate Tax as the major dependent variable in the smallest APC. This transform separates the two highest Tax values from the rest of the data — highlighting of these values show these cases have an almost exact correspondence in Ptratio, and have relatively high values of the pollution indicator, Noxsq. The smallest APC has identified a group of town districts (the districts explain the close correspondence between Tax and Ptratio) with high property taxes, and high pollution.

The interpretation through highlighting depends quintessentially on the additivity of the APC relationship : from this follows the linearity between transformed variables that leads to the highlighting method. The essence of the idea is that the APC-transforms guide selection of the cases for each variable, so that the nature of the dependence is clear.

8 CONCLUSION

As a natural extension of linear principal component methodology, additive principal componants have many potential fields of application. In psychometrics, where multiple correspondence analysis and non-linear principal components analysis have gained wide acceptance, the utility of APC techniques does not have to be argued. The use of the smallest APC as a method for detecting high dimensional structure in data, structure that cannot be easily detected even with sophisticated graphical tools, is a novel approach to multivariate data analysis.

The elegant Hilbert space theory underpinning the APC presents a strong case for this generalization of linear principal components. The characterization as an eigenfunction provides a large, well understood body of literature with which to approach theoretical considerations, and the task of estimation.

As an applied method, concern centers on two issues: the reliability of the estimates and the accessibility of the information it provides. Methods of assessing reliability, based on asymptotic results for eigenvalue estimation, are well known for the direct estimation methods; there is a lack of such results for general smoothing techniques.

The interpretation techniques we have presented are a first attempt at providing a readily accessible method for understanding the nonlinear dependencies of the smallest APC. This task of interpretation is not an easy one, clearly there are many approaches yet to be explored.

References

Benzecri, J. P. (1972). Sur l'Analyse des Tableaus Binaire Assoces a une Correspondance Multiple. Technical report, Universite Pierre et Marie Curie, Paris. Note Mimeo, Lab. Stat. Math.

Breiman, L. and Friedman, J. H. (1985). Estimating Optimal Transformations for Multiple Regression and Correlation. *Journal of the American Statistical Association*, 80:580 598.

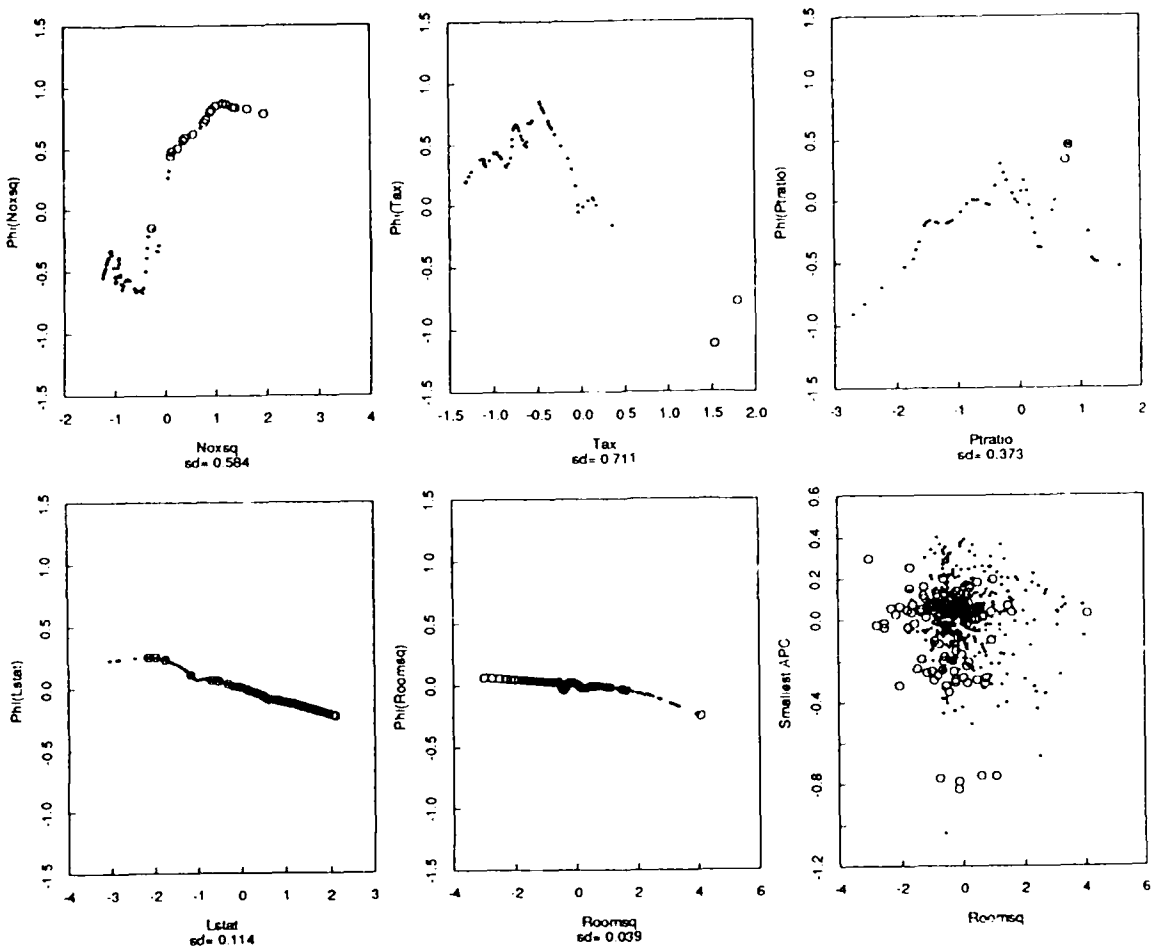


Figure 3: The smallest APC of the Boston Housing explanatory variables,**var** $\sum_i \phi_i(X_i) = 0.035$

De Leeuw, J. (1982a). Nonlinear Principal Components Analysis. *COMPSTAT*, pages 77–86.

De Leeuw, J. (1982b). Nonlinear Principal Components Analysis with B-splines. *Journal of Operations Research*, pages 379–394.

Gifi, A. (1981). *Non-linear Multivariate Analysis*. Leiden. Department of Data Theory.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. Wiley, New York.

Jorgens, K. (1970). *Linear Integral Operators*. Pitman Articles LTD, London.

Lebart, L., Morineau, A., and Warwick, K.(trans. Berry, E. M. (1984). *Multivariate Descriptive Statistical Analysis*. Wiley, New York.

Silvey, S. D. (1969). Multicollinearity and Imprecise Estimation. *J. Royal Stat. Soc. Ser. B*, 31:539–552.

STOCHASTIC TESTS OF FIT

P.W. Millar, University of California

0. Introduction.

This paper describes a method for using controlled randomization, coupled with computationally intensive methods, to resolve computational problems arising from a broad class of goodness of fit tests. Since the models whose fitness is being assessed here are, in general, non-parametric, a certain amount of care is necessary in choosing methods of numerical implementation. Issues surrounding the choice of method are discussed in sections 1,2. The main new result (cf., sec. 3) is a very general asymptotic representation theorem which, among other things, can be used to justify asymptotically both the methods proposed and the validity of bootstrap methods for calculating critical values from the approximating expressions.

1. Computational difficulties in certain tests of fit.

Let $x_1, x_2, \ldots, x_n$ be iid, R^d-valued random variables with unknown common distribution G. Let Θ be an index set, and $\{P_\theta, \theta \in \Theta\}$ a statistical model that can be either parametric, semiparametric, or non-parametric. An important question is to decide whether the model $\{P_\theta\}$ fits the data; more precisely, it is desired to test the null hypothesis that G belong to $\{P_\theta, \theta \in \Theta\}$.

A reasonable class of tests can be described as follows. Let B be a Banach space, T_θ a B-valued function of θ, and $\hat{T}_n$ a B-valued function of the data $x_1, \ldots, x_n$. If $|\cdot|$ denotes the norm of B, then a plausible goodness of fit statistic is

$$(1.1) \qquad \inf_{\theta \in \Theta} n^{1/2} |\hat{T}_n - T_\theta|,$$

the hypothesis being rejected for large values. While this recipe is reasonable in a large number of situations, the statistic (1.1) is incomputable, in general, for several reasons. To understand the computational difficulties surrounding (1.1), and to understand the computationally intensive substitutes for (1.1) which we develop later on, let us first look at several examples of statistics of the form (1.1).

Example 1.1: classical statistics. Let the data be real valued, let $\{P_\theta\}$ be a parametric family of probabilities on the line. Let T_θ be the cdf of P_θ: $T_\theta(t) = P_\theta\{x_1 \le t\}$ and let $\hat{T}_n$ be the empirical cdf of the data: $\hat{T}_n(t) = n^{-1} \Sigma I\{x_i \le t\}$. The Banach space B above can be taken to be the collection of real bounded functions f, with norm $|f| = \sup_t |f(t)|$ (i.e., $B = L_\infty(R^1)$). Then (1.1) becomes the usual *Kolmogorov-Smirnov goodness of fit statistic*, whose properties are well known if Θ consists of a single point.

To obtain a different classical statistic in this framework, let μ be a probability on R^1, and let B be the space of real functions f with norm $|f|^2 = \int f(s)^2 \mu(ds)$ (so now $B = L^2(\mu)$). Then, with T_θ, $\hat{T}_n$ as before, and the norm just given, (1.1) is a variant of the *Cramer-von Mises goodness of fit statistic*.

For the remaining examples, we shall, for convenience deal mainly with one particular space B. To describe it, let $S^d = \{s \in R^d : |s| = 1\}$ be the unit spherical shell of R^d. Define the halfspace $A(s,t)$ by

$$(1.2) \qquad A(s,t) = \{x \in R^d : x's \le t\}.$$

If P is any probability on R^d define $P(s,t)$ by

$$(1.3) \qquad P(s,t) = P\{A(s,t)\}.$$

By this means, the probability P is identified with an element of $L_\infty(S^d \times R^1)$, the Banach space of bounded continuous function on $S^d \times R^1$ with supremum norm. For d-dimensional data, half-spaces are the simplest class of sets which remains invariant under affine transformation and which separates measures: if $P(A) = Q(A)$ for all halfspaces A, then P and Q are identical as measures on the Borel sets. In certain problems, such as the location model below, halfspaces fit in with the structure of the model in an elegant way, leading to a much simpler analysis than one based, e.g., on lower left quadrants (cf (2.2c) below).

Example 1.2: parametric models. Let Θ be a subset of R^d, so that $\{P_\theta, \theta \in \Theta\}$ is a *parametric* model. A goodness of fit test of the form (1.1) is then

$$(1.4) \qquad \inf_{\theta \in \Theta} n^{1/2} \sup_{s,t} |\hat{T}_n(s,t) - T_\theta(s,t)|$$

where $T_\theta(s,t) = P_\theta\{A(s,t)\}$, and $\hat{T}_n(s,t) = \hat{P}_n\{A(s,t)\}$ and where $\hat{P}_n$ is the empirical measure of $x_1, \ldots, x_n$. One especially important case is when $\{P_\theta\}$ is the collection of normal distributions on R^d with unknown mean and covariance. Another important example, discussed briefly later on, concerns the Fisher distributions on the unit spherical shell in 3 dimensions. In this latter case, the supremum in (1.4) becomes $\sup_C n^{1/2} |\hat{P}_n(C) - P_\theta(C)|$ where C ranges over all *spherical caps on* S^3.

Example 1.3: symmetric location models on R^d. In R^d there are many notions of symmetry for random variables — for example,

(a) "simple symmetry": the rvX has the property that both X and $-X$ have the same distribution

(b) "isotropy": rvX has the property that X and γX have the same distribution for every orthogonal transformation γ. See Beran and Millar, 1988c, for further developments. Let $\mathbf{F}_0$ denote the collection of all probabilities on R^d that are "symmetric" according to, say, one of the possibilities just suggested. The $\mathbf{F}_0$ symmetry model asserts that for some *unknown* $\eta \in R^d$ and some *unknown* $F \in \mathbf{F}_0$, the *centered* data $x_1 - \eta, \ldots, x_n - \eta$ have distribution F. The parameter set Θ then consists of all pairs $\theta \equiv (\eta, F)$, $\eta \in R^d$ $F \in \mathbf{F}_0$; P_θ is the probability given by $P_\theta(A) = F(A - \eta)$, and (1.1) becomes

$$(1.5) \qquad \inf_{(\eta, F)} n^{1/2} \sup_{s,t} |\hat{T}_n(s,t) - T_\theta(s,t)|$$

where, as usual $T_\theta(s,t) = P_\theta(A(s,t))$, $\hat{T}_n(s,t) = \hat{P}_n(A(s,t))$.

Example 1.4: Logistic model. Here the data $x_1, \ldots, x_n$ takes the form $x_i = (y_i, z_i)$, where y_i takes on only the values 0,1 and where z_i are iid R^d-valued with distribution F. If $\beta = (\beta_0, \beta_1, \ldots, \beta_d) \in R^{d+1}$ and if F is a probability on R^d, then the logistic model asserts that $P\{y_i = 1 \mid z_i\} = P(\beta, z_i)$, where $\log P(\beta; z)[1 - P(\beta, z)]^{-1} = \beta_0 + \beta_1' z$, with $\beta_1 = (\beta_1, \ldots, \beta_d)$. Then $\theta = (\beta, F)$ where $\beta \in R^{d+1}$ and F is the probability governing the covariates $z_1, \ldots, z_d$. The family P_θ is given by $P_\theta\{y_i = 1, z_i \in A\} = \int_A P(\beta; z) F(dz)$. Define

$$T_\theta^i(s,t) = \int_{A(s,t):} [1 - P(\beta; z)]^{1-i} P(\beta; z)^i F(dz), \quad i = 0, 1$$

and $\hat{T}_n^i(s,t) = n^{-1} \Sigma I\{y_j = i, z_j \in A(s,t)\}$, $i = 0, 1$. Set $T_\theta = (T_\theta^0, T_\theta^1)$, $\hat{T}_n = (\hat{T}_\theta^0, \hat{T}_\theta^1)$, random elements in the Banach space $B \equiv L_\infty(S^d \times R') \times L_\infty(S^d \times R^1)$. With the Banach space just mentioned (the norm is the maximum of the norms of the factor spaces), the test statistic (1.1) for the logistic model is, with the *above* choices of $\hat{T}_n$, T_θ, B, given by (1.5)

With these examples behind us, we can now easily see the computational difficulties surrounding (1.1). First, the computation of $\inf_\theta$ will, for the supremum-type norms described above be intractable. In the location and logistics models, this calculation involves an infimum over an infinite dimensional collection of probabilities. Actually, this particular calculation is already intractable (for supremum norms) in the Gaussian and Fisher cases described in example 1.2. Second, the calculation for fixed θ of $|\hat{T}_n - T_\theta|$, is also intractable: in the examples 1.2-1.4, it involves computing the supremum over the collection of half-spaces in R^d. Finally, even the computation of $T_\theta(s,t) \equiv P_\theta(A(s,t))$ in examples 1.2-1.4 is typically intractable. In Gaussian parametric situations, the calculation is simple, because of properties of the normal distribution; on the other hand, for the Fisher distributions on S^3, the calculation of $T_\theta(s,t)$ (the mass given a particular "spherical cap" by a particular Fisher distribution) is intractable and must be obtained by approximation. Equally difficult computational difficulties can arise in the logistic and location models.

This paper describes very general, computationally intensive methods which can successfully confront the numerical intractability of (1.1). These methods involve (a) replacing the parameter set Θ by a (random) subset Θ_n (b) replacing the norm $|\cdot|$ of B by a random norm $|\cdot|_n$ and (c) replacing the functional T_θ by an approximating (random) functional $T_\theta^{(n)}$. The computationally feasible replacement of (1.1) takes the form

$$(1.6) \qquad \min_{\theta \in \Theta_n} n^{1/2} |\hat{T}_n - T_\theta^{(n)}|_n.$$

Because of the possible infinite dimensionality of Θ, the choices of $|\ |_n$, Θ_n must be made with some care; issues surrounding these choices are discussed in section 2.

2. Stochastic Methods.

This section discusses issues surrounding the choice of Θ_n, $T_\theta^{(n)}$, and $|\cdot|_n$ in the formula (1.6)

(2.1). Search of Θ. Henceforth, let us call the set $\Theta_n \subset \Theta$ used to construct (1.6), a *search set* for Θ; if Θ_n is random, we call it a *stochastic* search set. Throughout the rest of the paper, Θ is assumed to be a subset of a normed space.

Search method (a): sieves on Θ. Although the statistic (1.1) is generally incomputable, it is often *theoretically* intractable as well, because either the differentiability hypothesis of standard minimum distance theory may fail on Θ, or else the non-singularity hypothesis fails; for example differentiability problems arise in the location model and non-singularity fails for certain regression models (cf. Millar 1982). (These concepts are defined in section 3). It may be possible, however, to find (non-random) subsets $\Theta_n \uparrow \Theta$, such that these hypotheses hold on Θ_n sufficiently well that an asymptotic analysis may proceed, albeit with technical complications. The difficulties are reminiscent of those found in maximum likelihood estimation on infinite dimensional Θ (cf. Grenander, 1981; Geman, Huang, 1982) where the Θ_n so introduced are called "sieves".

The theoretical difficulties have a counterpart in sound intuition: it is undesirable to "over fit" the model relative to the data at hand. While sieve methods are of considerable theoretical interest, they frequently leave, in the situation of goodness of fit statistics like (1.1), a computational problem as bad as the original one

Search method (b): simple searches of Θ. A different modification of (1.1) is to replace $\inf_\Theta$ by a minimum over a *finite* subset $\Theta_n \subset \Theta$.

How should a finite search set be chosen? The set Θ is infinite dimensional, perhaps bounded, but it may not be precompact. In this case it would not be possible to construct a finite ε-grid over Θ. Even if Θ were compact, and thus there exists such a grid, actual construction of it could be formidable, except in very special cases. For example, construction of an ε-grid over all the probabilities on the unit ball of R^d appears to be intractable; such a difficulty can arise in both location and logistic models. Finally, construction of ε-grids is, in general, a much more ambitious undertaking than the one required to provide a decent approximation to $\inf_\Theta$.

Another suggestion might be to construct iid Θ-valued random variables $Y_1, \ldots, Y_{j_n}$, and take Θ_n to consist of the values of this sample. Except in special cases, there may be difficulty in carrying out this construction in a computationally feasible way.

A more fundamental difficulty centres on the fact that if P_{θ_0} is the actual data distribution, then typically $\inf_{\theta \in \Theta} |\hat{T}_n - T_\theta|$ achieves its minimum within a ball of radius $cn^{-1/2}$ about θ_0 (see section 3). On the other hand it can be shown (Millar, 1988) that for many bounded, infinite dimensional Θ, the Y_i-search, $1 \le i \le j_n$, will in general miss this crucial $n^{-1/2}$-ball with positive probability, no matter how fast $j_n \uparrow \infty$. More precisely, for *any* sequence $\{j_n\}$, there exists θ_0 and a sequence $\{a_n\}$, $a_n \gg n^{-1/2}$, such that $\underline{\lim} P\{|Y_i - \theta_0| > a_n$ for some $i \le j_n\} > 0$.

Search method (c): local stochastic search. A more

promising approach, which is justified by the result of section 3, depends upon the fact that, if θ_0 is the 'true' parameter, then the minimizing point of $n^{1/2}|\hat{T}_n - T_\theta|$ occurs within a neighborhood of diameter $n^{-1/2}$ about θ_0. Thus search methods which stray outside such neighborhoods waste time searching unimportant parts of Θ. One way to capitalize on this property is to suppose that there exist estimates $\hat{\theta}_n = \hat{\theta}_n(\underline{x}_n)$ of $\theta \in \Theta$, *with values in* Θ, such that, whenever $|\theta_n - \theta_0| \le cn^{-1/2}$, $\theta_n \in \Theta$:

(2.1) $\quad n^{1/2}|\hat{\theta}_n - \theta_n|$ is tight under P_{θ_n}

(We then call the estimators $\hat{\theta}_n$ "$n^{1/2}$ consistent"). In many nonparametric applications, such estimates are known: an example is given below. Next, let $\underline{x}_i^* = (x_{i1}^*, \ldots, x_{in}^*)$, $1 \le i \le j_n$ be j_n independent bootstrap samples of size n drawn from $P_{\hat{\theta}_n}$; cf. Efron, 1979. Define a random search set Θ_n by

(2.2) $\quad \Theta_n = \{\hat{\theta}_n(\underline{x}_n), \hat{\theta}_n(\underline{x}_1^*), \ldots, \hat{\theta}_n(\underline{x}_{j_n}^*)\}$

The search set (2.2) may be rewritten as $\Theta_n = \{\hat{\theta}_n(\underline{x}_n), \hat{\theta}_n(\underline{x}_n) + Y_1^* n^{-1/2}, \ldots, \hat{\theta}_n(\underline{x}_n) + Y_{j_n}^* n^{-1/2}\}$ where the Y_i^*, $1 \le i \le j_n$, are *conditionally* iid, given $\underline{x}_n$; here $Y_i^* = [\hat{\theta}_n(\underline{x}_i^*) - \hat{\theta}_n(\underline{x}_n)]n^{1/2}$. In our applications, Θ is typically *not open*. Therefore, the structurally simpler local search set $\{\hat{\theta}_n(\underline{x}_n), \hat{\theta}_n(\underline{x}_n) + Y_1 n^{-1/2}, \ldots, \hat{\theta}_n(\underline{x}_n) + Y_{j_n} n^{-1/2}\}$, where $Y_1, Y_2, \ldots, Y_{j_n}$ are iid, cannot be used here since there is no guarantee that $\hat{\theta}_n + Y_i n^{-1/2} \in \Theta$. Such difficulties arise in both the location and logistic models. On the other hand, this approach can be made to work in classical parametric problems, provided Θ is open; see also J.-P. Kreis, 1987, for another interesting case (involving time series) where such an iid local search is quite effective. Obviously, when the structurally simpler search just mentioned can be justified, it might be preferred because of the greater freedom in the method for simulating $Y_1, \ldots, Y_{j_n}$.

Example 2.1: Location model. What does the local stochastic search amount to in this case? Here is one possibility. Recall that here the parameter θ is $\theta \equiv (\eta, F)$, where η is the center of symmetry and F is the "symmetric" underlying distribution. An easy choice of estimate $\hat{\theta}_n \equiv (\hat{\eta}_n, \hat{F}_n)$, which satisfies the condition (2.1) begins by taking $\hat{\eta}_n$ to be an α-trimmed mean (coordinatewise trimming will do). Next, let $\hat{H}_n$ be the empirical measure of the centered data $x_1 - \hat{\eta}_n, \ldots, x_n - \hat{\eta}_n$, and let $\hat{F}_n$ be the "symmetrization" of $\hat{H}_n$. What $\hat{F}_n$ actually is depends on the exact definition of symmetry, but in the two illustrations given in Example 1.3, the definition of $\hat{F}_n$ is intuitively clear. For example, in the isotropic case, $\hat{F}_n$ puts the uniform distribution of weight n^{-1} on each of the spherical shells $\{x \in R^d: |x| = |x_i - \hat{\eta}_n|, 1 \le i \le n\}$. The stochastic search set Θ_n then consists of bootstrap replicas of $\hat{\theta}_n \equiv (\hat{\eta}_n, \hat{F}_n)$. This choice of $\hat{\theta}_n$ will be a $n^{-1/2}$ consistent estimator of $\theta = (\eta, F)$ with values in Θ.

(2.2). Stochastic norms. In this subsection we explain some methods for replacing $|\cdot|$ in (1.1) by a computable approximation $|\cdot|_n$, as in (1.6). Our procedure involves the notion of a random norm, a concept that, it turns out, has had a long history in statistical methods. For a probability space **X** and a linear space B, a *stochastic norm* $|\ \ |_n$ is a map from $\mathbf{X} \times B$ to R^1 such that for each $x \in \mathbf{X}$, $b \to |b|_n(\underline{x})$ is a pseudo-norm on B. Here are several examples of stochastic norms.

(2.2a). Kolmogorov distance.

Let $\underline{x}_n = (x_1, \ldots, x_n)$ be iid real random variables with continuous c.d.f. F, and empirical cdf $\hat{F}_n$. Let B denote the space of bounded, real, right continuous function on R^1 with left limits. A stochastic norm $|\ \ |_n$ on B is then given by $|b|_n = \max\{\max_{i \le n} |b(x_i)|, \max_{i \le n} |b(x_i-)|\}$, $b \in B$. The Kolmogorov distance between F, $\hat{F}_n$ is then given by the stochastic norm $|F - \hat{F}_n|_n$.

(2.2b). Cramer von Mises discrepancy. Let $\underline{x} = (x_1, \ldots, x_n)$, F, $\hat{F}_n$ be as in subsection (2.2a). The Cramer-von Mises discrepancy is $[\int [\hat{F}_n(t) - F(t)]^2 dF(t)]^{1/2}$. For fixed F this defines the $L^2(F)$ norm on $(\hat{F}_n - F)$. It is asymptotically equivalent to $[\int [\hat{F}_n(t) - F(t)]^2 d\hat{F}_n(t)]^{1/2}$, a stochastic norm $|\ \ |_n$ on the difference $\hat{F}_n - F$. The stochastic norm $|\ \ |_n$ is given by $|f|_n^2 \equiv \int f(t)^2 d\hat{F}_n(t)$; that is $L^2(\hat{F}_n)$ replaces the norm of $L^2(F)$.

(2.2c). Stochastic norms based on quadrants. For $t \in R^d$, let K(t) denote the lower left "quadrant" with corner at t: $K(t) = \{u \in R^d: u_i \le t_i, 1 \le i \le d\}$, where $u = (u_1, \ldots u_d)$, $t = (t_1, \ldots, t_d)$. Write P(t) for $P\{K(t)\}$, thus identifying P with an element of $B \equiv L_\infty(R^d)$. The quadrant metric between two probabilities P, Q is then $\sup_t |P(t) - Q(t)|$

There are several ways to replace the quadrant metric with a more computable approximation.

(i) Let $\underline{t} = (t_1, \ldots, t_{k_n})$ be a vector iid N(0, I) random variables on R^d. A simple stochastic norm $|\ \ |_n$ on $L_\infty(R^d)$ is then $|b|_n = \max_{i \le k_n} |b(t_i)|$, $b \in L_\infty$.

(ii) Let $x_1, \ldots, x_n$ be iid, R^d-valued, with empirical measure $\hat{P}_n$. Let C be the smallest cube in R^d containing the data points $x_1, \ldots, x_n$. Pave C with $(p_n)^d$ cubes of equal size; let n_i be the number of data points in cube i; draw $(n_i/n)k_n$ points uniformly from cube i. Then a stochastic norm on $L_\infty(R^d)$ is given by the maximum of $|b(t)|$, $b \in L_\infty$, over the points t just drawn. This data based stochastic norm will more nearly approximate the L_∞ norm of $\hat{P}_n - P$ than will the stochastic norm of (i) above. On the other hand, it involves more computation.

(2.2d). Stochastic norms on half-spaces. Let P be a probability on R^d, and write $P(s,t) = P(A(s,t))$, for the half-space A(s,t), as explained in (1.3), so that P is an element of $L_\infty(S^d \times R^1)$. Let $s_1, \ldots, s_{k_n}$ be iid, uniformly distributed on S^d, and $t_1, \ldots, t_{k_n}$ be iid N(0,1). Then two possible stochastic norms on $B = L_\infty(S^d \times R^1)$ are: $\max_{i \le k_n} |b(s_i, t_i)|$, and $\max_{i \le k_n} \sup_t |b(s_i, t)|$, $b \in B$. If b is of the form $b = \hat{P}_n - P$ where $\hat{P}_n$ is the empirical of n iid observations from P, then the second of these two sto-

chastic norms is more data dependent and thus appears to give a better approximation to the true norm. Data dependent stochastic norms along the lines of the second example under (2.2c) are also possible. Finding the "best" data dependent stochastic norms for estimating $|\hat{P}_n - P|$ for the half-space metric is currently an interesting open problem. As regards simulation of $s_1, \ldots, s_{k_n}$ here, note that the following simple method works: simulate $g_1, \ldots, g_{k_n}$ iid standard Gaussian rv's on R^d, and set $s_i = g_i / |g_i|$ where $|g_i|$ is the Euclidean norm of g_i.

2.3. Stochastic functionals and a generic form of the test statistics. Evaluation of the test statistic (1.1) often entails, as the examples of section 1 make clear, evaluation of quantities such as $P_\theta(A)$ for various parameters θ and certain sets A, such as half-spaces and quadrants. As in the Fisher example (cf. sec 1.) such an evaluation may not be easy. Thus, one wishes to replace $P_\theta(A)$ by an approximation $P_\theta^{(n)}(A)$, or more generally, the T_θ in (1.1) by an approximation T_θ^n. While the main result of section 3 gives a very general approach to this approximation problem, the special examples listed in section 1 have, in fact, involved only two kinds of approximation. First, for given θ, one may estimate $P_\theta(A)$ by drawing h_n iid. variables from P_θ, obtaining an empirical measure $\hat{P}_\theta^{(n)}$ which provides the estimate $\hat{P}_\theta^{(n)}(A)$. Such a method has been used in the Fisher distribution problem described in section 1 (cf. Beran and Millar, 1986). Of course, in some cases it may be extremely difficult to carry out such a simulation. In the location problem on R^d, the relevant probabilities $P_\theta(A)$ can be represented as averages of simple measures, over a certain Haar measure (which depends on the notion of symmetry adopted). While such an averaging is uncomputable, in general, replacing this Haar measure by an appropriate empirical probability yields an effective approximation. No doubt, in some situations it may be also possible to replace $P_\theta(A)$ by an analytic approximation. While there are a great many statistics of the form (1.1), (1.6) the *paradigm case* underlying the specific examples of section 1 is

$$(2.3) \qquad \min_{1 \le i \le j_n} \max_{1 \le j \le k_n} \sup_t n^{1/2} |\hat{P}_n(A(s_j, t)) - P_{\theta_i^*}^{(n)}(A(s_j, t))|$$

where $\{s_i\}$, $A(s,t)$ are given in (2.2d), $\{\theta_i^*\}$ is a collection of j_n bootstrap replicas of a preliminary $n^{1/2}$-consistent estimator of θ (cf. search method (c)) and $P_\theta^{(n)}$ is, for each θ, the empirical of l_n iid. random variables drawn from P_θ. Notice that this latter approximation need be performed only for $\theta = \theta_i^*$, $1 \le i \le j_n$; thus the Monte Carlo for the approximating functionals $P^{(n)}$ will depend upon the outcome of the bootstrap samples which determine the local search set for θ. As usual $\hat{P}_n$ in (2.3) is the empirical measure of the data and $\{P_\theta\}$ is a possibly nonparametric statistical model.

(2.4). Variable Θ_n vis a vis variable $|\ |_n$. The test statistics suggested by (1.2) have the form $\min_{\theta \in \Theta_n} |A_\theta|_n$, where Θ_n is a variable subset of Θ and $|\ |_n$ is a pseudo norm depending also on n. Even if $\Theta_n \uparrow \Theta$ and $|\ |_n \to |\ |$ (this last denoting a norm), one cannot in general hope for convergence of $\min_{\Theta_n} |A_\theta|_n$, no matter how smooth A_θ may be. The difficulty can often be traced to the infinite dimensionality of Θ: analogous objects in the finite dimensional case typically converge (Beran and Millar, 1988a).

Here is a simple illustration of the difficulty.

Example. Let B be the linear space of all real sequences $b = (b_1, b_2, \ldots)$ such that $b_k = 0$ for all sufficiently large k. For $a, b \in B$ define $< a, b > = \Sigma a_i b_i$. Let e_i denote the members of B with 1 on the i^{th} co-ordinate and 0 elsewhere. Define $|b|_{k_n} = \max_{i \le k_n} |< b, e_i >|$ and $\Theta_{j_n} = \{e_1, \ldots, e_{j_n}\}$. Then $\min_{\theta \in \Theta_{j_n}} |\theta|_{k_n} = 0$ $j_n > k_n$, $= 1$, $k_n \ge j_n$; thus, without conditions on j_n, k_n there can be no convergence of $\min_{\theta \in \Theta_{j_n}} |\theta|_{k_n}$ as $n \to \infty$.

Despite the simplicity of this example, its basic moral carries over: if convergence of $\min_{\Theta_n} |A_\theta|_n$ is desired, then the "size" of the search set Θ_n should not be "too sophisticated" for the norm $|\ |_n$. Intuitively, one should not "over fit" the model relative to the measure of discrepancy $|\ |_n$. In particular, applications will often therefore require conditions on the relative size of the search set Θ_n and the sample size determining the stochastic norm $|\ |_n$.

2.5. Critical values. Critical values for test statistics (1.6) can often be obtained by bootstrap method. There are several valid ways to do this. A technique which entails significant computational savings is a "conditional bootstrap method", which for convenience we describe only for the paradigm statistic (2.3).

First, *fix* the random variables $\{s_j,\ 1 \le j \le k_n\}$ which determine the stochastic norm; *fix* the random variables $\{\theta_i^*,\ 1 \le i \le j_n\}$ which determine the local stochastic search of Θ, and *fix* the simulated estimates $P_{\theta_i^*}^{(n)}$. Next, draw m_n bootstrap samples $\underline{u}_1^*, \ldots, \underline{u}_{m_n}^*$, each of size n, from the fitted model $P_{\hat{\theta}_n(\underline{x}_n)}$; here $\hat{\theta}_n$ is the same $n^{1/2}$-consistent estimator used to generate the search set for Θ. It is assumed that the $\underline{u}_i^*$ are conditionally independent (given $\underline{x}_n$) of the bootstrap samples used to construct θ_i^*, and of the random variables used to construct $P_\theta^{(n)}$, and independent of $\{s_i\}$. Let $\hat{P}_{\underline{u}_l^*}(\underline{u}_i^*; \cdot)$ denote the empirical measure of $\underline{u}_i^*$, and let G_n^* denote the empirical cdf of $\min_{1 \le i \le j_n} \max_{1 \le j \le k_n} \sup_t n^{1/2} |\hat{P}_n(\underline{u}_l^*; A(s_i, t)) - P_{\theta_i^*}^{(n)}(A(s_i, t))|$, $1 \le l \le m_n$. Then under suitable conditions the quantiles of G_n^* give asymptotically valid critical values for the stochastic test statistic (2.3). A proof can be based on the asymptotic representation theorem in section 3, together with techniques developed in Beran and Millar, 1987. Other valid bootstrap methods could involve recalculating either (or both) the search set $\{\theta_i^*\}$ or the $\{s_i\}$ for each bootstrap sample $\underline{u}_l^*$, $1 \le l \le m_n$. The added computational burden is enormous and up to first order asymptotics, there is no gain over the "conditional" method. Whether or not there is any compensating extra "stability" in such methods is an interesting open question that requires second order asymptotic analysis.

3. Asymptotic representation theorem.

This section establishes, under suitable hypotheses, the asymptotic form of test statistics of the form (1.6). The formulation is sufficiently general so that the result applies to statistics based on sieves (cf. section 2), to the stochastic statistics illustrated by (2.3), as well as to a number of other possibilities. It can be used to show under supplementary hypotheses that approximations like (1.6) have an asymptotic form similar to that of (1.1). The triangular array formulation makes the result a convenient tool in establishing the asymptotic validity of bootstrap methods in the calculation of critical values (cf subsection (2.5)). Motivation for the particular formulation adopted here comes from the "paradigm" statistic (2.3), the logistic testing problem (cf. section 1) which is *not* of the form given in the paradigm, and the possibility of extending sieve methods from an MLE framework to a "minimum distance" framework.

Let X be a measure space, and $\underset{\sim}{x}_n = (x_1, \ldots, x_n)$ a vector of X valued random variables having joint distribution P_n. Note that the general formulation does not require that the x_i be independent. Let Θ be a subset of a normed linear space B_1, and let Θ_n, n = 1,2,... be subsets of Θ. The subsets Θ_n are allowed to be random. For each $\theta \in \Theta$, let T_θ be a functional defined on X (or on the n-fold product of X) with values in a possibly different normed space B_2. Let $\hat{T}_n = \hat{T}_n(\underset{\sim}{x}_n)$ be a B_2-valued statistic on X^n.

Let $|\ \ |_n$ be a (possibly random) pseudonorm on B_2. For each θ, n, let T_θ^n be a B_2-valued functional on Θ also possibly random. In many applications T_θ^n is an easily computable approximation to T_θ, based on Monte Carlo simulations. The construction of Θ_n, $|\ \ |_n$, T_θ^n may involve certain auxiliary randomization. Let Q_n be the probability governing the distribution of $\underset{\sim}{x}_n$ as well as these constructions.

Fix $\theta_0 \in \Theta$. The hypotheses are as follows.

(3.1). Identifiability. For each $\varepsilon > 0$, $c > 0$ there exists $\delta > 0$ such that

$$\lim_{n\to\infty} Q_n\{ \inf_{\substack{\theta\in\Theta_n \\ |\theta-\theta_0|>c}} |T_\theta^n - T_{\theta_0}^n|_n \geq \delta\} \geq 1-\varepsilon$$

(3.2). Differentiability. There is a continuous linear map $\dot{l}$: span $\Theta \to B_2$ such that, for every $\varepsilon > 0$, there exists δ such that

$$\lim_{n\to\infty} Q_n[\sup_{\substack{|\theta-\theta_0|\leq\delta \\ \theta\in\Theta_n}} \{|T_\theta - T_{\theta_0} - \dot{l}(\theta-\theta_0)|_n / |\theta-\theta_0|\} \leq \varepsilon] = 1.$$

(3.3). Non-singularity.

$$|\dot{l}(\theta-\theta_0)|_n \geq C_n|\theta-\theta_0| \quad \forall\theta\in\Theta_n$$

where C_n^{-1} is a tight sequence under Q_n.

(3.4). Consistency of $\hat{T}_n$: $n^{1/2}|\hat{T}_n - T_{\theta_0}|_n$ is tight under Q_n

(3.5). Approximation property:

$$\sup_{\theta\in\Theta_n} |T_\theta^n - T_\theta|_n n^{1/2} \to 0$$

in Q_n probability.

(3.6). Proximity of θ_0:

$$\inf_{\theta\in\Theta_n} n^{1/2}|T_\theta - T_{\theta_0}|_n \equiv A_n \text{ is } Q_n \text{ tight.}$$

Condition (3.5) says that T_θ^n approximates T_θ in a suitable fashion. In the case where T_θ^n is an empirical measure obtained by simulating iid observations from a probability $P_\theta \equiv T_\theta$, familiar exponential bounds on the empirical process (cf, e.g., Alexander, 1984) quickly yield simple conditions on the size of Θ_n vis a vis the number of simulations used to construct T_θ^n. See Beran and Millar, 1988a for a simple illustration. If no approximation T_θ^n to T_θ is needed, as in the case of certain problems involving multivariate normal distributions, then of course (3.5) is automatically satisfied. The proximity condition (3.6) ensures that the point θ_0 is not far from Θ_n. If Θ_n consists of bootstrap replicas of a preliminary estimate, and if θ_0 is the 'true' parameter, then (3.6) is automatic. In case Θ_n is a sieve, (3.6) imposes conditions on the speed with which Θ_n exhausts Θ. The other conditions are n-dependent variants of familiar conditions from the theory of minimum distance estimators. Roughly speaking, the effect of (3.1) is to ensure that the minimizing θ-points for (1.6) can be found eventually (as $n \to \infty$) in any given "ball" about the "true" parameter; (3.2), (3.3) ensure that said "ball" has a diameter of order $n^{1/2}$. It is considerations such as these that suggest the efficiency of the local asymptotic search of Θ described in section 2.

Theorem 1. Assume (3.1) - (3.6). Let θ_n be any sequence such that $n^{1/2}(\theta_n - \theta_0)$ is norm bounded in B_1. Let $W_n = n^{1/2}[\hat{T}_n - T_{\theta_0}]$. Then under Q_n

$$\inf_{\theta\in\Theta_n} n^{1/2}|\hat{T}_n - T_\theta^n| = \inf_{\theta\in\Theta_n} |W_n - \dot{l}(n^{1/2}(\theta-\theta_0))|_n + o_{Q_n}(1).$$

A novelty of this formulation is that T_θ^n, Θ_n, $|\cdot|_n$ all depend on n and can be random. Moreover, the parameter set, Θ_n is not assumed open, unlike classical developments. Nevertheless, despite these novelties, the proof can be accomplished by a somewhat complicated extension of the methods of Wolfowitz (1953), Pollard (1980), Bolthausen (1977), Millar (1985) and others.

Remark. (a) The identifiability condition can be replaced by:

$$Q_n\{\exists \text{ at least one } \theta\in\Theta_n \text{ such that } |\theta-\theta_0|>c\} \to 0$$

as $n \to \infty$. When Θ_n is a local stochastic search as in Section 2, it is often quite easy to write down an analytic condition for the above convergence.

(b) Differentiability may be replaced by the following "asymptotic" differentiability condition, which will be employed elsewhere. There exist constants K_1, K_2 such that

$$|T_\theta - T_{\theta_0} - \dot{l}(\theta-\theta_0)|_n \leq K_1 n^{-\delta}|\theta-\theta_0| + K_2 n^\alpha |\theta-\theta_0|^2 + o_p(|\theta-\theta_0|)$$

where $\delta > 0$ and $\alpha < 1/2$.

(c) The derivative $\mathbf{l}$ can depend on n, provided $\mathbf{l}_n$ replaces $\mathbf{l}$ everywhere in the complete theorem statement.

(d) In many applications W_n, defined in the theorem statement, converges to W, $\{n^{1/2}(\theta - \theta_n): \theta \in \Theta_n\}$ more or less approximates span Θ, and $|\ \ |_n \to |\ \ |$, as $n \to \infty$. One therefore expects that the left side in theorem 1 converges to $\inf_{\theta \in \text{span } \Theta} |W - \mathbf{l}(\theta)|$, which is the classical form of the limit. In particular applications this convergence can indeed be established; however, as the example in subsection (2.4) makes clear, one cannot, in the generality considered here, expect such convergence to happen, without supplementary conditions which may regulate the size of Θ_n and the strength of $|\ \ |_n$.

Proof. Because of the approximation property (3.5), it suffices to analyse $\inf_{\theta \in \Theta_n} n^{1/2} |\hat{T}_n - T_\theta|_n$. Since

$$\inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| > c}} |\hat{T}_n - T_\theta|_n \geq \inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| > c}} |T_\theta - T_{\theta_0}|_n - |W_n|_n n^{-1/2},$$

hypotheses (4.1) (4.4) show that, for any c, $\inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| > c}} |\hat{T}_n - T_\theta|$ remains bounded away from 0 in probability. On the other hand, $|\hat{T}_n - T_{\theta_0}|_n \to 0$; this, together with (3.6) shows that $\inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| \leq c}} |\hat{T}_n - T_\theta|_n \to 0$ and so $\inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| \leq c}} |\hat{T}_n - T_\theta|_n = \inf_{\theta \in \Theta_n} |\hat{T}_n - T_\theta|_n$ for any preselected $c > 0$.

Next let $d_n = (\max\{A_n, b\})[|W_n|_n \vee 1]C_n^{-1}$ where A_n, C_n come from (3.6), (3.3). Then $\{d_n\}$ is tight. Without loss of generality, assume $c_n \leq \frac{1}{4}(\|\mathbf{l}\| + 1)^{-1}$. Let N_n be the set of $\theta \in \Theta_n$ such that $|T_\theta - T_{\theta_0} - \mathbf{l}(\theta - \theta_0)| \leq 1/2\, C_n |\theta - \theta_0|$. By (3.2), (3.3) and (3.6), N_n is nonempty, and by (3.3), the preceding paragraph, and an elementary argument $\inf_{\theta \in \Theta_n} |\hat{T}_n - T_\theta|_n = \inf_{\theta \in N_n} |T_n - T_\theta|_n$. On the other hand, if $\theta \in N_n$, $|\hat{T}_n - T_\theta|_n \geq |\mathbf{l}(\theta - \theta_0)|_n - |T_\theta - T_{\theta_0}|_n - 1/2\, C_n |\theta - \theta_0| \geq 1/2\, C_n |\theta - \theta_0| - n^{-1/2}|W_n|_n$. Since $N_n \supset \{\theta \in \Theta_n: n^{1/2}|\theta - \theta_0| \leq d_n\}$, which is nonempty by (3.6) and the definition of d_n, the calculation in the preceding sentence implies

$$\inf_{\theta \in N_n} n^{1/2} |T_n - T_\theta|_n = \inf_{\substack{\theta \in \Theta_n \\ n^{1/2}|\theta - \theta_0| \leq d_n}} |\hat{T}_n - T_\theta| n^{1/2}$$

$$= \inf_{\substack{\theta \in \Theta_n \\ n^{1/2}|\theta - \theta_0| \leq d_n}} |W_n - \mathbf{l}(\theta - \theta_n) n^{1/2})|_n + o_{Q_n}(1).$$

Next, note that, with Q_n probability approaching 1,

$$\inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| \leq d_n n^{1/2}}} |W_n - \mathbf{l}(\theta - \theta_0) n^{1/2}|_n \leq (\|\mathbf{l}\| + 1) d_n.$$

On the other hand, by definition of d_n, if $|\theta - \theta_0| > d_n$, then, since $C_n \leq 1/4(\|\mathbf{l}\| + 1)^{-1}$:

$$|W_n - \mathbf{l}(n^{1/2}(\theta - \theta_0))|_n \geq |\mathbf{l}(n^{1/2}(\theta - \theta_0))| - |W_n|$$

$$\geq c_n^{-1} d_n - d_n$$

$$\geq 4(\|\mathbf{l}\| + 1) d_n - d_n \geq 3(\|\mathbf{l}\| + 1) d_n$$

so that $\inf_{\substack{\theta \in \Theta_n \\ |\theta - \theta_0| \leq d_n}} |W_n - \mathbf{l}(n^{1/2}(\theta - \theta_0))|_n$

$= \inf_{\theta \in \Theta_n} |W_n - \mathbf{l}(n^{1/2}(\theta - \theta_0))|_n$, q.e.d.

1. Research supported by National Science Foundation grant, DMS87-01426.

References

Alexander, K. (1984). Probability inequalities for empirical processes and a law of the iterated logarithm. *Ann. Prob.*, **12**, 1041-1057

Beran, R.J. and Millar, P.W. (1986). Confidence sets for a multivariate distribution. *Ann. Statist.*, **14**, 431-443.

—— (1987). Stochastic estimation and testing. *Ann. Statist.*, **15**, 1131-1154.

—— (1988a). A stochastic minimum distance test for multivariate parametric models. To appear, *Ann. Statist.*

—— (1988b). Tests of fit for logistic models. Tech. Report No. 160 Department of Statistics. University of California, Berkeley.

—— (1988c). Multivariate symmetry models. Tech. Report No. 159, Department of Statistics. University of California, Berkeley.

Bolthausen, E. (1977). Convergence in distribution of minimum distance estimators. *Metrika*, **24**, 215-227.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Ann. Statist.*, **7**, 1-26.

Geman, S. and Hwang, C.-R. (1982). Nonparametric maximum likelihood estimation by the method of sieves. *Ann. Statist.*, **10**, 401-414.

Grenander, U. (1981). Abstract Inference. Wiley, New York.

Kreis, J.-P. (1987). On stochastic estimation. Preprint.

Millar, P.W. (1985). A general approach to the optimality of minimum distance estimators. *Trans. Amer. Math. Soc.*, **286**, 377-418.

—— (1982). Optimal estimation of a general regression function. *Ann. Statist.*, **10**, 717-40.

—— (1988). On the coverage of a set by a random sample. Preprint.

Pollard, D. (1980). The minimum distance method of testing. *Metrika*, **27**, 43-70.

Wolfowitz, J. (1957). The minimum distance method. *Ann. Math. Statist.*, **28**, 75-88.

BOOTSTRAP INFERENCE FOR REPLICATED EXPERIMENTS

Walter Liggett, National Bureau of Standards, Gaithersburg, MD 20899

ABSTRACT

Inference methods valid for nonnormal error are proposed for experiments in which each design point is replicated three or more times. Differences between the replicates provide the data needed for a pooled estimate of the error density, and this density forms the basis for the bootstrap. The density estimator is specified for symmetric error in Liggett (Biometrika, 1988), and this symmetric estimator has been generalized to asymmetric error. In this paper, the application of this density estimator to designed experiments is considered. The lack-of-fit test is of particular interest. The extension of the density estimator to data requiring a blocking variable and to data with dispersion effects is discussed. The bootstrap based on this density estimator is shown to be valid for smaller sample sizes when the test statistics are robust. Estimation of the error density is illustrated with measurements replicated at different laboratories.

1. INTRODUCTION

In industrial experimentation, when the error properties are crucial to the inferences drawn, the possibility of nonnormal experimental error must be considered. One source of variability that is potentially nonnormal is the inhomogeneity of physical samples of bulk materials. For example, trace concentrations of a particulate substance in portions of a bulk material are usually nonnormal. Another potentially nonnormal source is material degradation that accelerates as it proceeds. Many corrosion processes have this property as does spoilage due to bacterial growth. Other potentially nonnormal sources are inherent in measurement procedures. Examples include loss of analyte during preparation of the physical sample, interfering peaks in spectra and chromatograms, aberrant results from the software that automatically locates peaks and measures their height or area, and inconsistent control of variation due to poor understanding of the sensitivities of the measurement procedure. This paper discusses a bootstrap method for obtaining valid inferences when the experimental error is nonnormal.

An approach to bootstrap inferences for replicated experiments is provided by the pooled error density estimator given by Liggett (1988, 1989). This estimator is based on the assumption that replicate measurements involve independent and identically-distributed realizations of the measurement error, the usual assumption in designed experiments. This assumption leads to a relationship between the error density and the densities of the first and second differences between error realizations. These latter densities can be estimated from differences between replicates. The computation of the error density is a fitting by weighted, nonlinear least squares.

Bootstrap inferences in regression can be based on other density estimators (Efron, 1982; Efron and Tibshirani, 1986). When each design point is replicated three or more times, a separate density estimate for each design point is an alternative to a pooled density estimate. In this case, each bootstrap repetition is obtained by separately sampling with replacement the measurements at each design point (Efron, 1982). When the number of replicates at each design point is small, this approach suffers from the discrepancies between the small-sample empirical distributions and the true error distributions. If a pooled density estimator can be justified, other pooled estimators might be chosen in place of the replicate-differences density estimator in Liggett (1988, 1989). A model of the true values at the design points can be fit to the data, and the residuals can be computed and combined to form a density estimate. These residuals might be obtained from separate location estimates for each design point or from a more restrictive model of the regression function. When the number of replicates is small, the location estimates for the design points are unstable, and the naive combination of residuals does not provide a completely adequate density estimate. The combination of residuals from a more restrictive regression model leads to an error density that depends on the design matrix. Accounting for this dependence in a lack-of-fit test seems difficult. The replicate-differences density estimator does not suffer from these problems and thus seems attractive.

Because of the possibility of nonnormal error, robust statistics are the proper choice for the desired inferences (Hampel, et al., 1986). The use of the bootstrap to find the distribution of robust statistics is the focus of our discussion. Thus, our interest is in robustness of validity for statistics with good robustness of efficiency. Hampel, et al. (1986) offer robust tests for linear models based on the asymptotic distribution of the test statistics. We propose to use the same test statistics but to replace the asymptotic distribution with the bootstrap.

Designs with three or more replicates at each design point have recently been recommended for applications in which dispersion effects are potentially important (Box, 1988 and the discussion). These designs are also appropriate for the density estimates given in Liggett (1988, 1989). The recommendation of designs with three or more replicates at each design point is a considerable change from the usual recommendation. For example, the number of replicates recommended for lack-of-fit tests may be as small as 5, a number too small for investigation of either dispersion effects or nonnormality. When the experimental error is dominated by a single error source, the error is often both nonnormal and has a variance that depends on the controllable factors in the experiment. Thus, dispersion effects must be considered when nonnormality is important, and conversely, nonnormality must be considered when dispersion effects are important. The inclusion of both nonnormality and dispersion effects in the analysis is often needed.

In this paper, three aspects of the application of the replicate-differences density estimator to designed experiments are considered. The first, which is discussed in Section 2, is the extension

of the density estimation method to the case in which the replicates of the experiment require a blocking variable and to the case in which dispersion effects are present. The second, which we discuss in section 3, is the effect of the choice of test statistic and experimental design on the validity of the bootstrap when the sample sizes are moderate. The third, which we discuss in Section 4, is the performance of the density estimator on a set of measurements with a variety of real-world imperfections.

2. DENSITY ESTIMATION

The model on which this paper is based differs from the usual model for designed experiments only in the omission of the assumption of normality. The jth replicate measurement at the kth design point is given by

$$y_{jk} = x_k^T\theta + \varepsilon_{jk} \quad (j = 1,\ldots,r_k;\ k = 1,\ldots,K) \quad (1)$$

where x_k^T is the row of the design matrix corresponding to the kth design point, θ is the vector of unknown parameters, ε_{jk} is the zero mean, independent and identically-distributed error, r_k is the number of replicates at the kth design point, and K is the number of design points.

As specified in Liggett (1988, 1989), the replicate-differences estimator of the density of ε_{jk} is based on the Hermite function expansion (Schwartz, 1967). Note that this orthogonal function expansion is different from the Edgeworth expansion. The Hermite functions can be defined by the recursion

$$\begin{aligned}\varphi_0(x) &= \pi^{-\frac{1}{4}} \exp(-x^2/2)\\ \varphi_1(x) &= 2^{\frac{1}{2}}\pi^{-\frac{1}{4}} x \exp(-x^2/2)\\ \varphi_q(x) &= (2/q)^{\frac{1}{2}} x\, \varphi_{q-1}(x) - \{(q-1)/q\}^{\frac{1}{2}} \varphi_{q-2}(x).\end{aligned} \quad (2)$$

To apply the Hermite function expansion to the measurements, we use a scale factor computed from the median of the absolute differences between replicate measurements

$$s_r = (0.6745)^{-1}(2)^{-\frac{1}{2}} \operatorname{median}_{(1\le j<j'\le r_k,\ k=1,\ldots,K)} |y_{jk} - y_{j'k}|. \quad (3)$$

Division by 0.6745 and $\sqrt{2}$ makes s_r an unbiased estimate of the error standard deviation in the normal case. We estimate the error density by fitting the functional form for the density given by

$$\tilde{p}(x) = (1/s_r) \sum_{q=0}^{Q} a_q \varphi_q\{(x+\alpha)/s_r\}, \quad (4)$$

where the parameter α is chosen so that the mean of $\tilde{p}$ is zero. If the error density is assumed to be symmetric as in Liggett (1988), then $\alpha = 0$ and only Hermite functions of even order are needed in the expansion.

The error density is estimated through its relation to the densities of the first and second differences between replicate measurements at the same design point. The estimates of these densities on which the fitting is based are given by

$$\hat{p}_d(x) = (\sqrt{2}s_r)^{-1} \sum_{q=0}^{Q} \hat{d}_{2q}\, \varphi_{2q}\{x/(\sqrt{2}s_r)\}, \quad (5)$$

$$\hat{d}_{2q} = \Big\{\sum_{k=1}^{K}(r_k-1)\Big\}^{-1} \sum_{k=1}^{K}\frac{2}{r_k} \times \sum_{j>j'} \varphi_{2q}\{(y_{jk} - y_{j'k})/(\sqrt{2}s_r)\}, \quad (6)$$

$$\hat{p}_c(x) = (\sqrt{6}s_r)^{-1} \sum_{q=0}^{3Q} \hat{c}_q\, \varphi_q\{x/(\sqrt{6}s_r)\}, \quad (7)$$

$$\hat{c}_q = \Big\{\sum_{k=1}^{K}(r_k-1)\Big\}^{-1} \sum_{k=1}^{K}\frac{2}{r_k(r_k-2)} \times \sum_{h}\sum_{j>j',\ j,j'\neq h} \varphi_q\{(2y_{hk} - y_{jk} - y_{j'k})/(\sqrt{6}s_r)\}. \quad (8)$$

Equations (6) and (8) show explicitly how the replicate differences enter the error density estimation. As specified in Liggett (1988, 1989), the fitting is accomplished by a weighted nonlinear least squares algorithm. Approaches to avoiding negative values of the density estimate are presented in Liggett (1989).

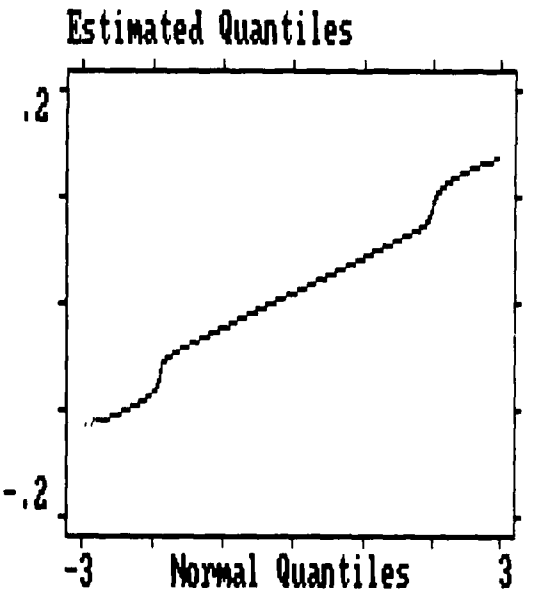


Figure 1. Error density estimated from Quinlan's cable-shrinkage experiment (Box, 1988).

An experiment with 4 replicates at each of 16 design points was performed by Quinlan (Box, 1988). Quinlan included many replicates to facilitate the analysis of dispersion effects. The reanalysis of Quinlan's data by Box and the discussants of Box's paper (Box, 1988) show that the dispersion effects are not particularly strong. Ignoring the dispersion effects, we can obtain an error density estimate from Quinlan's data. This estimate is interesting despite the violation of the assumption of identically-distributed error that underlies the replicate-

differences density estimate. Figure 1 shows the error density estimate based on the assumption of symmetry. Figure 1 is a quantile-quantile plot of the estimated error density versus the normal density. We see that the center of the estimated density looks very much like the normal, but that the tails of the estimated density are somewhat thicker than the normal.

The tails of the error in Quinlan's data can be investigated further by means of a *half-normal* probability plot of the absolute differences between replicate measurements. This probability plot, which is shown in Figure 2, does not appear to be perfectly straight. Rather, this figure suggests, as does Figure 1, that the density of the absolute differences has a tail somewhat thicker than the normal.

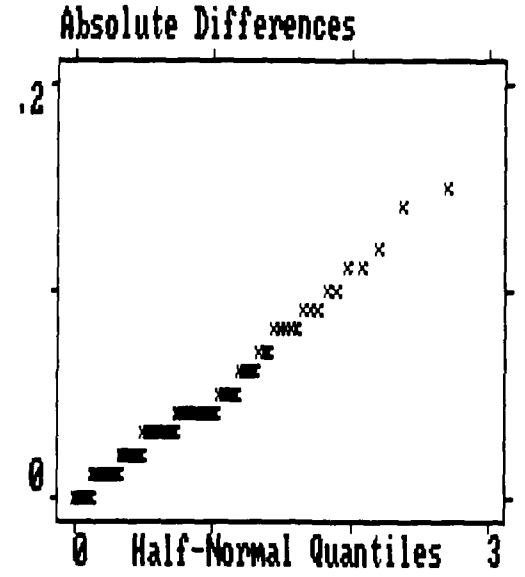


Figure 2. Replicate differences from Quinlan's cable-shrinkage experiment (Box, 1988).

The existence of dispersion effects is a reasonable explanation for the thick tails apparent in Figures 1 and 2 since dispersion effects give a set of error realizations that appear to arise from normal scale mixture if the dependence on the experimental factors is ignored. Thus, Figures 1 and 2 do not provide any important clarification of Quinlan's data. Consideration of Quinlan's data in this paper is intended to link replicate-differences density estimation with the designs adopted for the analysis of dispersion. This link raises the question of how both analyses can be combined.

Box (1988) mentions that Quinlan's experiment was run in the "split-plot" mode and that the experiment may involve two error components only one of which is reflected in the replicates. One way to mitigate this problem is to measure all the design points once, then measure all the design points a second time, and continue to repeat this as many times as necessary. If this procedure were to be followed, then we would likely have to include a blocking variable, that is, to replace ε_{jk} in (1) with $\delta_j + \varepsilon_{jk}$ to obtain an adequate model of the measurements. To estimate the error density, we would first have to estimate the values of δ_j. Various ways to estimate the δ_j suggest themselves. In the present context, estimation of the δ_j by maximizing $\hat{d}_0$ seems interesting since this method can be thought of as choosing the δ_j to make the error look as normal as possible. Differentiation of $\hat{d}_0$ shows that the resulting estimate of δ_j is an M-estimate with redescending ψ function.

The most general way to combine nonnormality and dispersion effects in an error model is to allow the error density to depend in some unknown way on the controllable factors. Such a model would limit the pooling that could be done in the estimation of the error densities and would thus require a very large number of replicate measurements. One way to limit the number of measurements required is to assume that the dispersion effects only involve the scale of the error so that after the replicate-differences have been corrected for the scale effects, the error density can be estimated by pooling all the corrected differences. Let the error term in (1) be given by $\sigma_k \varepsilon_{jk}$, where the dependence of σ_k on k, the design point, can be modeled by a function with fewer unknown parameters than K, the number of design points. We propose to estimate σ_k and then correct the replicate-differences using this estimate. The estimators of dispersion effects suggested by Box and the discussants (Box, 1988) may be appropriate. A robust estimator for the dispersion effects might be better.

3. BOOTSTRAP INFERENCE

Bootstrap inference consists of finding the distribution of a statistic by computing realizations of the statistic from independent samples drawn from an estimated density. In this paper, we focus on statistics for testing lack of fit, which is an important inference in designed experiments. Other important inferences in designed experiments involve confidence intervals for the differences between points on the response surface and confidence intervals for the values of σ. The validity of bootstrap inference depends on both the accuracy of the density estimate and on the characteristics of the statistic, which in turn, depend on the experimental design. In this section, we consider how the choice of design and statistic affect the validity of bootstrap inference based on the replicate-differences density estimate.

The percentiles of the replicate-differences density estimate have larger bias and larger standard deviation in the tails than at the center of the distribution. This suggests that this density estimate will provide accurate percentiles for a location estimate robust against stretched-tail error even when the percentiles of the density estimate itself seem inaccurate. This principle can be illustrated with the median of three. If the error density and error distribution are given by $p(x)$ and $F(x)$, respectively, then the density of the median of three independent error realizations is given by $6F(1-F)p$. The factor $6F(1-F)$ downweights the tails of p.

To provide some specific insight into the validity of the bootstrap based on the replicate-differences density estimate, we consider an example in which the sample size is small for the

purpose of density estimation and the error distribution is quite asymmetric. We consider 3 replicates at each of 20 design points and error distributed as χ^2 with 3 degrees of freedom. In the Hermite function expansion of the error density (4), we let Q = 10. The mean and standard deviation of the percentiles obtained in 100 trials are

	Error Density			Median of Three		
Prob.	True*	Mean	Std Dev	True*	Mean	Std Dev
0.01	-2.89	-3.81	1.02	-2.60	-3.02	0.55
0.05	-2.65	-3.12	0.57	-2.26	-2.38	0.47
0.10	-2.42	-2.65	0.50	-2.01	-2.00	0.46
0.25	-1.79	-1.71	0.46	-1.46	-1.33	0.44
0.50	-0.63	-0.45	0.35	-0.63	-0.45	0.35
0.75	1.11	1.17	0.37	0.46	0.62	0.30
0.90	3.25	3.16	0.83	1.69	1.67	0.49
0.95	4.81	4.75	1.25	2.56	2.48	0.71
0.99	8.34	6.56	1.84	4.45	4.43	1.19

*The mean of the χ^2 with 3 df is subtracted.

Dividing the standard deviations by 10 to obtain standard errors of the means, we see that the density estimates are biased near the center and in the tails. In the tails, both the bias and the standard deviation are smaller for the median of three than for a single error realization. Thus, the replicate-differences density estimate clearly provides more accurate results for the median of three. These trials suggest that even for the median of 3, the design, 3 replicates at 20 design points, and the error density, χ^2 with 3 degrees of freedom, may not lead to an adequately stable density estimate. In an application of the replicate-differences density estimate, the effect of the stability of the density estimate on the desired inferences should be investigated by Monte Carlo experiment.

To test lack of fit, we specialize the τ-tests discussed by Hampel, et al. (1986). A lack-of-fit test is a comparison of the fit of the model of interest with the fit of the most general model that can be estimated, namely, a location estimate for each design point based on just the measurements at that design point. Let u_1, u_2, ... be variables that specify the factor levels in the experimental design, and let $x^T\theta$ be a K-term polynomial in these factors, $x^T = (1, u_1, u_2, u_1^2, u_1u_2, \ldots)$. We can choose such a polynomial with the property that all the elements of θ can be estimated and the property that the model we wish to test is given by setting $\theta_m = 0$ for elements in x not in the model. Let x_k be the value of x at the design point k. Hampel, et al. (1986, p. 346) offer a test based on

$$\Gamma(\theta) = \sum_{k=1}^{K} \sum_{j=1}^{r_k} \tau(x_k, (y_{jk} - x_k^T\theta)/\sigma), \qquad (9)$$

where the function τ is chosen based on the desired robustness properties and σ is a scale parameter that must be estimated. Our notation differs from that in Hampel, et al. (1986) in obvious ways. Hampel, et al. (1986) propose the statistic

$$S_n^2 = \frac{2}{K-h}\frac{1}{n}\left[\min_\theta\{\Gamma(\theta)|\theta_m=0 \text{ terms not in model}\} - \min_\theta\{\Gamma(\theta)\}\right], \qquad (10)$$

where $n = \Sigma r_k$, the total number of measurements, and h is the number of terms in the model to be tested. Hampel, et al. (1986) give the asymptotic distribution of S_n^2 and propose that tests be carried out on the basis of this distribution. As an alternative, we propose to use bootstrap samples from the replicate-differences density estimate to determine whether an observed value of S_n^2 is statistically significant. Work is needed to determine the situations under which this proposal has major advantages.

Consider the major issues involved in the choice of the function τ. For lack-of-fit tests derived from τ-tests, the issue of bounded influence does not arise and thus, the choice of τ is simplified. For lack-of-fit tests, the minimization of $\Gamma(\theta)$ under the full model is simply fitting a separate location estimate to each design point. Thus, no design point has higher influence than any other and no choice of τ provides dependence on the design point. Since no design point is downweighted, a single design point might cause rejection of the fit of the model. This behavior is what is usually expected of lack-of-fit tests.

Another issue is whether to choose a τ with a redescending ψ function. On one hand, a redescending ψ function provides superior performance when severe outliers are present. On the other hand, with a redescending ψ function, the minimum of $\Gamma(\theta)$ under the model might be such that a design point is completely ignored in both the estimate of θ under the model and the value of $\Gamma(\theta)$ that appears in the statistic S_n^2. Thus, the test might lead to acceptance of the fit of the model even though all the replicates at one design point have very large residuals. Belief that all the measurements at one design point can be rejected as outliers does not seem reasonable. One way out of this dilemma is to estimate θ using a τ that has a redescending ψ function but to avoid a redescending ψ function in the $\Gamma(\theta)$ chosen for the statistic S_n^2. In other words, in testing lack of fit, the τ-test, which is based on the same τ for estimation and testing, might be generalized to different τ functions for estimation and testing.

The validity of a bootstrap based on the replicate-differences density estimate also depends on the choice of τ. As we have already noted, the choice of a robust test is important for validity. Moreover, whether the ψ function is redescending may be have an effect on validity. Roughly speaking, in the case of stretched-tail error, the replicate-differences density estimate tends to have shorter tails than the true density estimate. Thus, for a redescending ψ function, the bootstrap samples may have fewer observations that have no influence than samples from the true distribution would have. For a non-redescending ψ function such as Huber's, the location of outliers beyond a certain point makes no difference.

The design of the experiment also has a bearing on validity. For some designs, the evidence for lack of fit comes from only one design point. An example is a centerpoint that has been added to a

two-level factorial design. This is the case in which an analysis of the experiment based on normality may not be saved by the central limit theorem. This is also the case in which validity depends on the percentiles of the distribution of the location estimate for a single design point. For other designs, the evidence of lack of fit is spread over many design points. In this case, the bootstrap should be valid over a broader range of error distributions and sample sizes.

4. APPLICATION

In this section, we consider a set of kinematic viscosity measurements made on re-refined oil. The set of measurements consists of 3 measurements on each of 65 samples of re-refined oil (Weeks, et al, 1983). The measurements are of the kinematic viscosity at 100 °C. In this application, the actual variability of the samples, which is of interest, must be distinguished from the measurement error, which is not normally distributed. Each oil sample was measured by three different laboratories. However, since the same standard measurement method was used by each laboratory and since the interlaboratory bias was corrected on the basis of reference sample measurements, the model given by (1) is plausible. The non-normality of the measurement error is manifested in two ways. In the set, 5 measurements differ markedly from the corresponding measurements by the other two laboratories. Even without these outliers, a half-normal probability plot of the differences between measurements on the same sample shows evidence of an error density that has a longer tail than the normal. Consider a statistic for comparison of the variability of oil samples from various sources. An appropriate statistic might be computed from the medians of the three measurements on each oil sample. The contribution of the measurement error to this statistic can be assessed by means of a bootstrap based on the error density estimate.

An estimate of the error density of the kinematic viscosity measurements was computed. Before considering this estimate itself, we consider two diagnostic quantile-quantile plots, a plot of the empirical distribution of the absolute differences between replicates on the same oil sample versus the distribution of these differences obtained from the estimated density, and a plot of the empirical distribution of the second differences versus their distribution as obtained from the estimated density. The plot of the absolute differences in Figure 3 shows that the estimated density fits the data well except for the 10 differences that involve the 5 outliers. Similarly, the plot of the second differences in Figure 4 shows that the estimated density fits the data well except for the 15 second differences that involve the 5 outliers, four of which are high, and one low. Clearly, the estimated density does not account for the extreme values of the differences. These figures contain a warning about the interpretation of the estimated density. Also, these figures suggest that a better error density estimate might be obtained by increasing Q so that the error density estimate can better represent the tails.

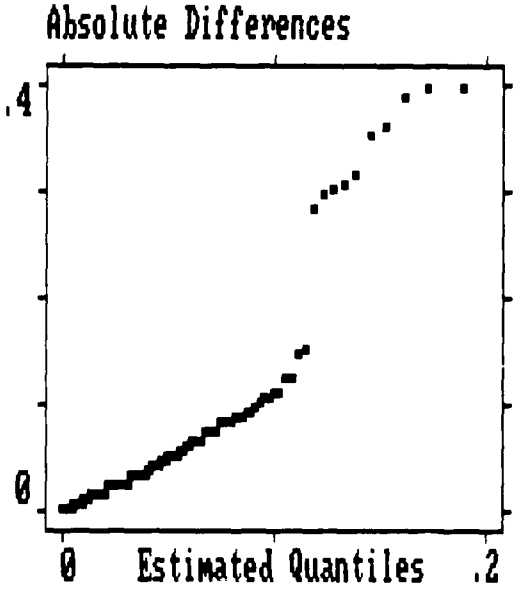


Figure 3. First differences between replicate kinematic viscosity measurements, empirical versus estimated distribution.

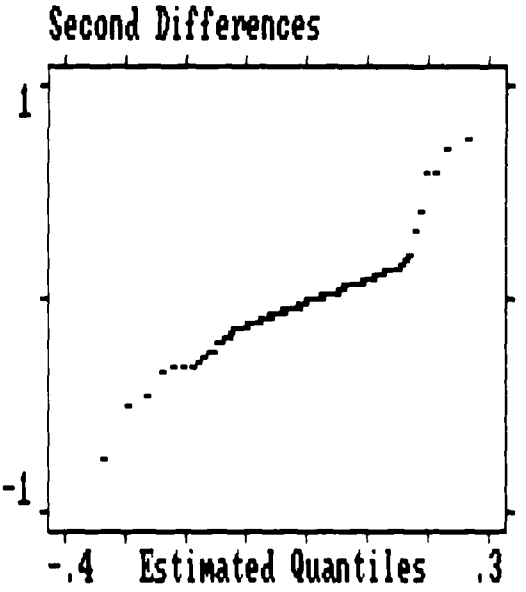


Figure 4. Second differences between replicate kinematic viscosity measurements, empirical versus estimated distribution.

Figure 5 shows a quantile-quantile plot of the estimated distribution versus the normal distribution. The error density appears to be negatively skewed, but this conclusion must be tempered by the results in Figures 3 and 4. One way to check the effect of the outliers is to remove them from the data set and re-estimate the error density. The result of this is shown in Figure 6. Both Figures 5 and 6 show the same basic shape for the error density, a negative skewness. Thus, we conclude that our error density estimator largely ignored the 5 outliers.

This behavior can be explained by the factor $\exp(-x^2/2)$ in the Hermite functions that appear in (6) and (8). These viscosity measurements suggest the appropriateness of our error density estimator for bootstrap inference for robust estimates.

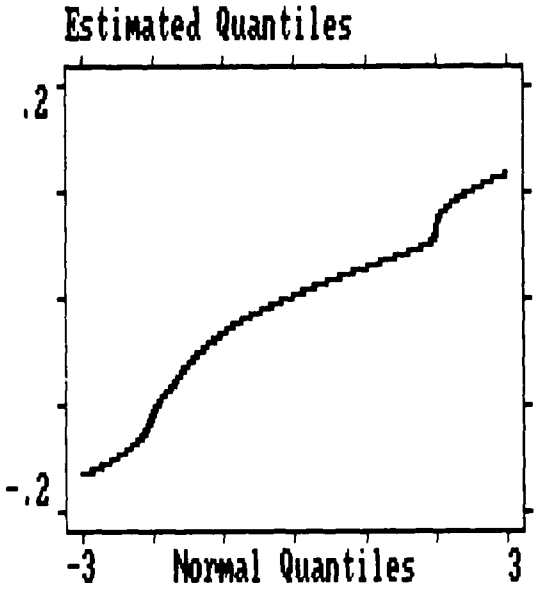


Figure 5. Error density estimated from kinematic viscosity measurements.

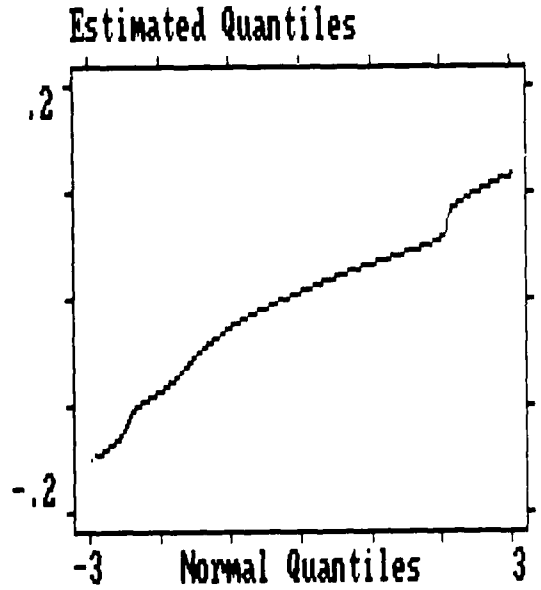


Figure 6. Error density estimated from kinematic viscosity measurements remaining after outlier removal.

REFERENCES

BOX, G. (1988). Signal-to-noise ratios, performance criteria, and transformations (with discussion). Technometrics 30, 1-40.

EFRON, B. (1982). The Jackknife, the Bootstrap, and Other Resampling Plans. Philadelphia: Society of Industrial and Applied Mathematics.

EFRON, B. & TIBSHIRANI, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statist. Science 1, 54-77.

HAMPEL, F.R., RONCHETTI, E.M., ROUSSEEUW, P.J., & STAHEL, W.A. (1986). Robust Statistics The Approach Based on Influence Functions. New York: Wiley.

LIGGETT, W.S. (1988). Estimation of the error probability density from replicate measurements on several items. Biometrika 75.

LIGGETT, W.S. (1989). Estimation of an asymmetrical density from several small samples. Submitted for publication.

SCHWARTZ, S.C. (1967). Estimation of probability density by orthogonal series. Ann. Math. Statist. 38, 1261-1265.

WEEKS, S.J., BECKER, D.A., & HSU, S.M. (1983). ASTM/NBS Basestock Consistency Study Data, U.S. National Bureau of Standards Special Publication 661. Washington: U.S. Government Printing Office.

REGRESSION STRATEGIES

David Brownstone*, University of California, Irvine

INTRODUCTION

Almost all statistics and econometrics texts contain strong admonitions against sequential estimation (or "data mining"). These admonitions are as effective as those against teen-age sex and drug abuse. Applied econometricians ignore the textbook warnings and use sequential strategies because they believe that they yield better estimates. In spite of considerable efforts, theoretical statisticians have been unable to analyze the sampling properties of these strategies under realistic conditions (see Judge and Bock (1978) and Judge (1984))[1]. This study solves this problem by using the bootstrap (see Efron(1982) and Efron and Gong(1982)) to compute the sampling distribution of different estimation strategies.

This paper examines the sampling properties of simple multiple regression estimation strategies based on variable and outlier deletion. With only small deviations from a model with orthogonal regressors and normally distributed errors there are substantial biases in the standard errors and t-statistics reported at the last stage of these simple strategies. Since the bootstrap is only asymptotically valid, the results presented here are based on Monte Carlo repetitions from known error distributions to eliminate the confounding effects of possible small sample biases. However, for all of the designs considered in this paper, the small sample biases in the nonparametric bootstrap are negligible. The key conclusion of this work is the necessity of completely specifying the estimation strategy and then bootstrapping it to get consistent estimates of the sampling distribution.

The bootstrap technique works by generating artificial data samples and computing the estimator for each sample. This technique has been used to derive small sample properties of estimators for autoregressive linear models by Freedman and Peters (1984) and for Nested Logit models by Brownstone and Small(1988). Independently, and more recently, Kipnis (1987) and Veall(1987) have used the bootstrap to examine the effects of various estimation strategies in linear regression models. Kipnis looks at the strategy of choosing the subset of variables to maximize R^2 in a model with orthogonal regressors, but he does not report the properties of individual coefficient estimators. Veall considers these properties for a stepwise regression strategy applied to an empirical example. Although his results are qualitatively similar to those in this study, it is impossible to disentangle the effects of possible model misspecification from the biases caused by the estimation strategy.

Since most applied econometricians use some sequential estimation strategy but only report the biased t-statistics and standard errors from the last stage, this study concentrates on examining the size of these biases for a number of known models. The design of the experiments concentrates on isolating the effects of multicollinearity among the regressors. The biases reported here are caused solely by the use of sequential estimation strategies. This study is not designed to explore exactly when the biases will be large, but rather to show how pervasive large biases are and suggest a methodology for removing them. In particular, there are always large negative biases in the standard errors estimated from the last round of even simple sequential procedures. With moderate collinearity, these biases are frequently greater than 100 per cent.

Although it would be interesting to further isolate the causes of the biases from sequential estimation, there is clearly large potential gain from designing better estimation strategies. The bootstrap methods described in this study could then be used to generate consistent estimates of the sampling distribution of these strategies.

EXPERIMENTAL DESIGN

The experiments are designed to investigate the sampling properties of two estimation strategies, Ordinary Least Squares (OLS) and Sequential OLS, with and without deletion of outliers and influential observations. The Sequential OLS (abbreviated by SEQ) procedure used in this study consists of first

estimating the full model by OLS, deleting all variables (except the first two) with absolute T–statistics less than 2, and finally estimating the restricted model by OLS. The T–statistics for this procedure are calculated from the usual OLS formulas at the second stage.

There are 7 regressors and 100 observations in each data set. The regressors and true parameter values are initially generated as independent draws from a unit Normal distribution. Each set of independent regressors is then transformed into 5 increasingly collinear data sets. The most collinear data had condition numbers[2] of approximately 135. These data are examples of strongly collinear data, but such data are quite commonly encountered in applied econometric work. The highest bivariate correlation between any two regressors in any of the experiments is 0.9.

For each of the four experiments, 100 independent draws of the regressors and true parameters were made. For each of these draws, and each of the 5 collinear data sets based on them, 300 independent "dependent variables" were generated according to

$$Y = X\beta + \epsilon,$$

where X and β are fixed, and the ϵ are independent unit Normal random variables or, for two of the experiments, independent unit Normal contaminated with 10% independent draws from a Normal distribution with mean 0 and variance 100[3]. The sampling distribution of each of the estimation strategies is then estimated from the sample of estimates over the 300 bootstrap repetitions[4]. All of the results presented in this study pertain to the estimation of the coefficient of the second of the two variables which were always kept in the regressions.

Each experiment considers the SEQ and OLS estimation strategies for 100 independent draws of the true parameter values and 6 increasingly collinear regressor matrices. In two of the experiments, outliers and influential observations were deleted before the estimation strategies were calculated. Outliers are those observations with standardized residuals[5] greater than 2 Influential observations are those with "hat" values greater than 0.14. These measures, and choice of cutoff values, are fully described in Belsley, Kuh, and Welsch (1980).

The four experiments used in this study are chosen to investigate commonly used variable and outlier deletion strategies across a wide range of realistic model settings. The experiments are:

RUN1: OLS and Sequential OLS (SEQ) are compared where the dependent variable is uncontaminated (e.g. the ϵs are all draws from a unit Normal distribution).

RUN2: OLS and SEQ are compared where the dependent variable is contaminated (e.g. 10% of the ϵs are drawn from a Normal distribution with variance equal 100).

RUN3: After first removing outliers and influential observations, SEQ and OLS are compared where the dependent variable is uncontaminated.

RUN4: After first removing outliers and influential observations, SEQ and OLS are compared where the dependent variable is contaminated.

100 different basic models (initial X matrix and β values) are used for each experiment since there is no *a priori* reason to expect convergence to anything over these repetitions. The purpose of these repetitions is to investigate the behavior of the estimation strategies across different models and to insure that the results are not artifacts of some peculiar X or β values. Finally, since the matrices for transforming the basic independent X matrices into collinear regressors were fixed across all of the runs, the repetitions also induce small variations in collinearity around the experimental design values.

RESULTS

The results of the four experimental runs are presented as percentiles across the 100 basic data repetitions in Tables 1 – 4 . The numeric suffixes on the row labels refer to the degree of collinearity in the X matrix. The suffix 1 refers to the basic independent data set, and higher suffixes correspond to the increasingly collinear transforms of these data. The rows

TABLE 1: RUN1 RESULTS
Uncontaminated Errors, No Outlier Deletion

Percentiles	5	25	50	75	95
COND1	1.36	1.46	1.52	1.58	1.66
EFF1	-4.16	-1.26	-0.19	0.01	1.15
BIOLS1	-5.57	-1.62	1.80	3.84	6.56
BISEQ1	-4.69	-0.39	2.14	4.39	7.61
COND2	5.16	6.27	6.72	7.25	8.11
EFF2	-44.96	-16.56	-6.31	-0.69	5.45
BIOLS2	-5.76	-1.41	1.04	3.49	6.34
BISEQ2	-3.30	2.86	6.44	15.03	33.20
COND3	10.34	12.66	13.65	14.85	17.00
EFF3	-69.69	-39.66	-20.66	-5.02	12.65
BIOLS3	-5.99	-1.81	1.45	3.63	7.69
BISEQ3	-2.12	9.30	23.45	38.09	72.92
COND4	29.35	36.79	40.72	44.29	51.00
EFF4	-120.04	-66.68	-30.98	21.96	89.89
BIOLS4	-7.20	-1.17	0.77	3.76	8.13
BISEQ4	5.26	38.09	63.30	86.85	117.39
COND5	58.17	73.31	81.51	88.15	102.25
EFF5	-132.83	-66.24	7.60	60.75	111.65
BIOLS5	-7.29	-2.36	0.28	2.73	6.37
BISEQ5	11.37	32.72	63.39	96.11	128.30
COND6	96.78	121.72	135.88	146.74	170.45
EFF6	-125.98	-41.93	27.43	75.78	137.30
BIOLS6	-5.38	-1.79	0.25	3.18	7.54
BISEQ6	11.59	29.46	59.25	94.29	141.82

TABLE 2: RUN2 RESULTS
Contaminated Errors, No Outlier Deletion

Percentiles	5	25	50	75	95
COND1	1.39	1.48	1.53	1.61	1.67
EFF1	-8.35	-3.92	-2.35	-0.03	3.74
BIOLS1	4.22	8.22	10.92	14.01	19.10
BISEQ1	4.44	9.21	12.09	14.31	19.33
COND2	4.60	6.19	6.91	7.41	8.29
EFF2	-52.82	-28.19	-13.30	-2.16	22.15
BIOLS2	-0.08	4.83	8.23	12.11	17.77
BISEQ2	5.75	15.39	24.05	35.97	51.11
COND3	9.15	12.46	14.17	15.19	17.17
EFF3	-80.17	-29.88	1.44	23.25	64.68
BIOLS3	-2.42	5.13	9.57	12.99	17.25
BISEQ3	14.96	27.53	38.16	51.54	66.36
COND4	27.48	37.45	41.24	45.59	52.45
EFF4	-57.50	24.80	57.65	96.01	134.18
BIOLS4	-0.31	5.79	9.00	13.25	19.81
BISEQ4	13.37	24.22	41.12	60.09	109.46
COND5	54.10	74.58	82.41	91.83	105.18
EFF5	-65.79	50.05	89.00	121.69	148.44
BIOLS5	-2.89	6.20	10.44	13.22	18.97
BISEQ5	8.34	23.00	36.97	59.85	120.41
COND6	89.81	124.32	137.34	153.21	175.36
EFF6	-43.33	56.68	95.34	128.77	151.01
BIOLS6	-0.28	6.01	9.75	13.25	20.23
BISEQ6	14.80	22.82	38.61	60.75	108.46

labelled "COND" give the condition number for the X matrices.

The other three rows in each group give the properties of the estimated second regressor coefficient (recall that the first two regressors were always included). The row labelled "EFF" gives the percentage improvement in the mean square estimation (MSE) error of SEQ versus OLS: positive values imply that SEQ is a better estimator. Note that the MSE here is measured relative to the true parameter value used to generate the dependent variables.

The remaining two rows (prefixes BIOLS and BISEQ) in the Tables give the percentage bias in the T-statistics for the two estimation strategies. These biases are computed by comparing the average of the standard OLS T-statistics from the last stage regression over the 300 bootstrap repetitions with:

$$\hat{T} = \frac{\Sigma\, b_i}{\sqrt{\Sigma(b_i - \bar{b})^2}}$$

where b_i denotes the estimate of the second element of β at the ith bootstrap repetition and $\bar{b}$ is the sample mean of the b_i. Since Tables 1–4 are based on a Monte Carlo study with the error vectors drawn from their known true distributions, $\hat{T}$ converges to the true T-statistic as the number of bootstrap repetitions gets large.

If the error vectors are drawn from the empirical distribution of the residuals from a regression using all of the regressors, the resulting $\bar{b}$ would be Efron's

TABLE 3: RUN3 RESULTS
Uncontaminated Errors With Outlier Deletion

Percentiles	5	25	50	75	95
COND1	1.36	1.46	1.52	1.59	1.70
EFF1	-4.49	-1.30	-0.28	0.01	1.10
BIOLS1	12.41	17.07	20.83	23.80	29.18
BISEQ1	13.49	17.57	21.54	24.51	29.77
COND2	5.00	6.18	6.92	7.51	8.44
EFF2	-34.37	-17.22	-4.16	-0.72	3.29
BIOLS2	12.71	17.23	19.60	23.15	26.02
BISEQ2	12.52	21.60	25.78	34.27	47.85
COND3	9.98	12.65	14.09	15.48	17.29
EFF3	-62.82	-31.62	-13.52	-3.43	23.82
BIOLS3	13.71	16.80	21.26	24.35	27.51
BISEQ3	17.54	26.95	41.97	56.69	103.73
COND4	28.38	37.05	41.91	46.80	52.97
EFF4	-109.17	-48.01	-16.94	24.76	63.45
BIOLS4	12.20	17.39	21.52	24.67	28.99
BISEQ4	34.41	53.72	77.91	101.03	150.48
COND5	56.78	74.05	84.27	93.65	106.37
EFF5	-105.51	-24.69	15.17	55.82	88.88
BIOLS5	14.61	18.04	20.36	22.94	26.94
BISEQ5	28.99	53.99	79.04	99.73	162.33
COND6	94.66	123.43	140.65	156.13	177.41
EFF6	-104.89	-15.20	42.87	76.36	96.63
BIOLS6	12.88	17.48	21.56	24.10	28.15
BISEQ6	31.22	51.77	77.37	95.24	150.99

TABLE 4: RUN4 RESULTS
Contaminated Errors With Outlier Deletion

Percentiles	5	25	50	75	95
COND1	1.38	1.46	1.50	1.61	1.72
EFF1	-7.69	-3.31	-1.13	-0.02	2.29
BIOLS1	2.01	7.78	10.61	14.16	20.15
BISEQ1	3.52	8.73	11.88	14.93	19.65
COND2	5.52	6.13	6.77	7.46	8.21
EFF2	-42.97	-22.55	-10.90	-3.44	16.02
BIOLS2	3.87	7.75	11.36	15.01	19.46
BISEQ2	12.29	17.19	22.82	28.56	47.48
COND3	10.94	12.39	13.76	15.46	16.99
EFF3	-79.91	-42.80	-18.56	-3.41	32.56
BIOLS3	-0.20	7.59	11.50	14.43	20.16
BISEQ3	12.41	25.47	35.28	54.37	76.82
COND4	31.35	36.65	41.38	45.92	51.32
EFF4	-101.95	-39.32	-2.31	40.63	79.29
BIOLS4	-1.20	7.87	11.28	13.74	19.80
BISEQ4	23.64	42.65	64.45	86.20	132.02
COND5	62.35	73.44	82.88	91.75	103.33
EFF5	-81.06	-20.41	36.51	79.96	120.77
BIOLS5	0.63	6.50	10.41	13.19	18.13
BISEQ5	18.83	35.70	60.52	84.14	135.69
COND6	103.80	122.40	138.21	152.95	172.52
EFF6	-103.36	2.11	50.34	91.57	135.16
BIOLS6	-0.61	6.89	10.24	13.56	20.03
BISEQ6	12.95	29.32	54.59	87.97	124.89

nonparametric bootstrap estimator. Since all of the models considered here satisfy the Gauss–Markov assumptions, Efron's (1982) results show that $\hat{T}$ converges to an unbiased test statistic for the hypothesis that Plim $\bar{b} = 0$ as the number of bootstrap repetitions gets large. Note that these estimators do not require knowledge of the true model so that they can be applied in real situations. The small sample accuracy of this bootstrap estimator was checked by rerunning all of the experiments with the error vectors drawn from their empirical distributions. In all cases the results are almost identical to the Monte Carlo results in Tables 1–4, thus justifying the use of the nonparametric bootstrap at least for these experimental designs.

Table 1 gives the results of the first experiment, RUN1, with uncontaminated errors and no outlier deletion. As textbook theory predicts, there are no biases or differences between OLS and SEQ if the regressors are independent (corresponding to the suffix "1" in the tables). The OLS T–statistics are also unbiased since they are Uniform Minimum Variance Unbiased estimators in this situation. However, with even mild collinearity, there are substantial efficiency differences between OLS and SEQ. More striking are the increasingly large positive biases in the T–statistics for the SEQ strategy: these biases average 60 per cent and frequently exceed 100 per cent. With multicollinearity it is possible to considerably improve estimation efficiency using the SEQ strategy, but the resulting T–statistics will certainly be overestimates.

FIGURE 1: BIAS IN SEQ T-STATISTIC, RUN1

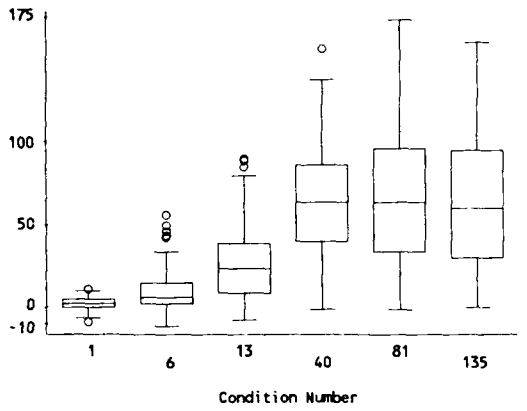
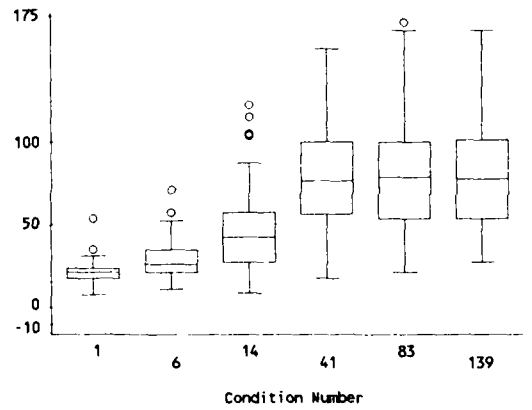


FIGURE 2: BIAS IN SEQ T-STATISTIC, RUN3



Notes for Figures 1 and 2:

These figures show box plots of the percentage bias in the T-statistics for the SEQ estimation strategy. Box plots, originally designed by Tukey, are common tools in exploratory data analysis. They are described in textbooks like Kitchens (1987). The upper part of the box is at the 75th percentile, the line in the middle of the box is at the median (50th percentile) and the lower part of the box is at the 25th percentile. The upper whisker is at the "upper adjacent value," which is the closest observation to the 75th percentile + 1.5 x the Interquartile Range (the 75th - 25th percentile). The open circles denote outliers, which are any observations past the adjacent values. If the data followed a Normal distribution, then we would only expect to see .7 of these outliers per box in any of the plots.

RUN2 considers the same estimation strategies in a case where 10 per cent of the errors are contaminated. The results, shown in Table 2, are similar to those in RUN1. One difference is that now some of T-statistics for OLS for collinear regressors are positively biased, although these biases are much smaller than the positive biases in the T-statistics for SEQ. In addition there are now clearer efficiency gains to using the SEQ strategy as collinearity increases.

RUN3 has the same data generation process as RUN1, but the OLS and SEQ strategies are modified by first removing outlying and/or influential observations. As expected from the properties of the data generating process, approximately 10 percent of the observations are removed in each replication. The efficiency comparisons between SEQ and OLS are similar to those for RUN1. Note that now the T-statistics for both OLS and SEQ are biased even for orthogonal regressors. The magnitude of this bias increases with collinearity for SEQ, but remains constant for OLS.

RUN4 compares the same estimation strategies as RUN3 for data generated with contaminated errors. Since there are now some serious outliers to be removed, the estimators should perform better than in RUN3. Although the biases in the T-statistics are lower than in RUN3 for both estimation strategies, the biases for the SEQ strategy are still very large for highly collinear regressors.

One common feature of all the results presented in Tables 1 - 4 is the large variation in almost all the measures across the different data designs. Figures 1 and 2 graphically show the bias in the T-statistics for the SEQ strategy in RUN1 and RUN2. The large magnitudes of the efficiency differences and biases clearly suggest that there is large potential gain from developing better estimation strategies.

CONCLUSIONS

The simulations show the dangers in using the results of common estimation strategies for hypothesis testing. Although this study only considers simple linear regression models, I expect the qualitative conclusions to hold for more complex econometric models. The

methodology used here can easily be applied to analyzing any estimation strategy for any well–specified model[6].

This study also demonstrates the feasibility of using the bootstrap to generate consistent estimates of the sampling distribution of estimation strategies for multiple regression models. The large differences in estimation efficiency between the OLS and SEQ strategy show that there is large potential gain from designing better strategies. Even if one only uses OLS, there are still substantial biases in the T–statistics when there are outliers and/or influential observations.

Although it would be interesting to explore the conditions where SEQ works well in these experiments, theoretical work (Belsley et. al. (1980) and Judge and Bock (1978)) suggest that these conditions will depend on unknown parameters. Bootstrapping allows consistent estimation of the sampling distribution of any sequential procedure, which allows comparisons to be made for each model and data set.

REFERENCES

Belsley, D., E. Kuh, and R. Welsch (1980), Regression Diagnostics, John Wiley and Sons, New York.

Brownstone, D. and K. Small (1988), "Efficient Estimation of Nested Logit Models," Journal of Business and Economic Statistics, forthcoming.

Efron, B. (1982), The Jackknife, Bootstrap and Other Resampling Plans, SIAM, Philadelphia.

Efron, B. and G. Gong (1982), "A Leisurely Look at the Bootstrap, the Jackknife, and Cross–Validation," The American Statistician, V.37, Pp. 36 – 48.

Freedman, D. A. and S. C. Peters (1984), "Bootstrapping a Regression Equation: Some Empirical Results," Journal of the American Statistical Association, V. 79, Pp. 97 – 106.

Judge, G. G. (ed.) (1984), "Pre–Test and Stein–Rule Estimators: Some New Results," Journal of Econometrics, V. 25, No. 1/2

Judge,G. G. and M. E. Bock (1978), The Statistical Implications of Pre–Test and Stein–Rule Estimators in Econometrics, North–Holland, Amsterdam.

Kipnis, V. (1987), "Model Selection and Predictive Assessment in Multiple Regression," Dept. of Economics, University of Southern California, Mimeo.

Kitchens, L. J. (1987), Exploring Statistics: A Modern Introduction, West Publishing Co., Saint Paul, Minnesota.

Veall, M. (1987), "Bootstrapping the Process of Model Selection: An Econometric Example," Dept. of Economics, McMaster University, Mimeo.

NOTES

*Financial support from a UCI Academic Senate Faculty Fellowship grant is gratefully acknowledged. The author also wishes to thank Ami Glazer, Ken Small, and Carole Uhlaner for helpful comments. Of course, they bear no responsibility for any remaining flaws.

[1]Although some of the simplest experiments reported here could be analyzed using analytic techniques, the experiments involving outlier and influential observation can not.

[2]The condition number is defined to be the ratio of the largest and the smallest eigenvalue of the moment matrix (X'X) of the independent variables. See Belsley, Kuh, and Welsch (1980) for further information and justification for this as a measure of multicollinearity.

[3]A randomly chosen 10% of the ϵs are independent normal random variables with mean 0 and variance 100, and the rest are independent unit normally distributed. The 10 per cent figure is based on the fundamental "law" of survey statistics which states that 10 per cent of any data set is garbage.

[4]In a number of cases, 600 bootstrap repititions were run as a convergence check. There were no significant changes in the results after 200 repititions.

[5]As suggested in Belsley, Kuh, and Welsch (1980), these standardized residuals were computed by first excluding the observation in question from the standard error calculations.

[6]Computational costs may become prohibitive in more complex settings. All simulations for this study were performed on an 8Mhz. PC/AT clone with a total running time of 120 hours.

DATA SENSITIVITY COMPUTATION FOR MAXIMUM LIKELIHOOD ESTIMATION

Daniel C. Chin
The Johns Hopkins University Applied Physics Laboratory

ABSTRACT

This paper presents a computational procedure and the numerical results for studying the effects of outliers or other anomalous data on maximum likelihood estimates. This procedure is based on a first order approximation relying on the implicit function theorem. The numerical results of this paper are given for a multivariate signal-plus-noise problem with independent non-identically distributed noise terms. These numerical studies will illustrate the procedure.

1. INTRODUCTION

This paper presents an efficient method of determining the sensitivity of maximum likelihood estimates (MLEs) to the data used in calculating the estimates. This method is much more efficient than a standard simulation that would involve several recomputations of MLE and is useful in predicting the effect of outliers or anomalous data on the estimate.

Maximum likelihood estimation is widely used in statistical analysis. It is found in estimating the instrumentation error for a guidance system or a navigation system, in the geodetic parameter errors for an earth model, and in orbit determination for satellites. In addition, the MLE is also utilized in macroeconomic modeling, biometrics problems, and education. Because of sophisticated equipment and the complications of the real world, the dimension of a MLE problem can be very large; therefore to have over a hundred parameters in a single case is very common. Since the MLE has no closed form solutions, it is very costly to find a MLE. To find many MLEs for data sensitivity studies is even harder. Therefore, it is worth the effort to develop a method which can approximate MLEs in a quick and accurate fashion. This method is different from the sampling techniques discussed in Iman [1980].

Section 2 will present the approximation method named the First-Order IFAP, or Implicit Function Approximation. The general IFAP theory can be found in Spall [1986]. Section 3 presents numerical studies on the signal-plus-noise problems, and Section 4 is a brief conclusion.

2. AN APPLICATION OF THE IFT

The First-Order IFAP contains the first two terms of the Taylor expansion around the existing MLE using the implicit function theory (IFT). Since only the "first-order" will be discussed, this hyphenated word will be omitted. The nonlinear estimation for nonlocal sensitivity can be found in Kalaba [1986]. As discussed in Spall [1985], IFAP pertains to an approximation framework of the form handled by a parameter estimator, that is, from data x_1, $x_2,\ldots,x_n$, $x_i \sim N(\mu, \Sigma + P_i)$, IFAP can be used to gain insight into the properties of the maximum likelihood (ML) estimate, $\hat{\beta}$, of the vector of unique and relevant parameters, β, in μ and Σ. This study demonstrates how the current software can be used to study the influence of anomalies or outliers within the set of x_i's on the estimate $\hat{\beta}$.

Assume that $\hat{\beta}$ is found as the root of the score equation, i.e.,

$$\hat{\beta} = \left\{\beta: \frac{\partial L}{\partial \beta} = 0\right\} \qquad (1)$$

where L represents the log-likelihood function. Since (1) involves a term like $\partial L/\partial \Sigma = 0$ and since it may be that $\Sigma \ngeq 0$ satisfies $\partial L/\partial \Sigma = 0$, IFAP is not necessarily working with a constrained (positive semidefinite) estimate of Σ. A further restriction of the current IFAP formulation is that all P_i's are assumed to exist (i.e., $(P_i^{-1})^{-1}$ exists). We believe that a modification of IFAP to accommodate either the square-root formulation (i.e., the procedure for ensuring $\hat{\Sigma} \geq 0$) or the so-called information formulation, which relies on P_i^{-1} instead of the nonexistent P_i, would be fairly straightforward.

Given an observed set of data, $x^* \equiv (x_1^{*T}, x_2^{*T}, \ldots, x_N^{*T})^T$ and $P_1, P_2, \ldots, P_N$, the present software computes quantities related to the first-order expansion.

$$\hat{\beta}(x|x^*) = \hat{\beta}(x^*) + T_1^*(x - x^*)$$

where $T_1^* = [d\hat{\beta}/dx^T]_{x^*,\hat{\beta}^*}$ is computed using the implicit function theorem and $\hat{\beta}^* = \hat{\beta}(x^*)$. The various quantities computed from $\hat{\beta}$ and T_1^* include several unit-free, normalized measures of sensitivity which will be discussed in greater detail in the next section.

3. SIGNAL-PLUS-NOISE EXAMPLES

There are three subsections in this section. Subsection I will describe how that data was generated. Subsection II shall demonstrate the accuracy of the approximations. Subsection III uses two examples to show the IFAP results.

I. Introduction

A series of numerical results were generated to evaluate the IFAP methodology and its software. These results use the same set of input

$$\{x_i^*, P_i\} \text{ for } i = 1, 2, \ldots, 25$$

where

$x_i \in R^{15}$

$P_i = A_i A_i^T$ for $i = 1, 2, \ldots, 25$

A_i is a 15 x 30 matrix with its elements, $\{a_{ij}\}$, generated randomly using the uniform distribution over (-1, 1).

x_i^* is generated using normal distribution $N(0, P_i + \Sigma)$

$$\Sigma = \begin{bmatrix} 15^2 & & & 0 \\ & 15^2 & & \\ & & \ddots & \\ 0 & & & 15^2 \end{bmatrix}$$

A nonconstrained scoring algorithm is used to get a maximum likelihood (ML) estimate, $\hat{\beta}^* = \hat{\beta}(x^*)$, as the baseline value. The parameters of β include all the means $(\mu_1, \mu_2, \ldots, \mu_{15})$ and the nonzero part of the following covariance matrix:

$$\begin{bmatrix} \Sigma_{1,1} & & & & & & & & \\ \Sigma_{2,1} & \Sigma_{2,2} & & & & & & 0 & \\ \vdots & \vdots & \ddots & & & & & & \\ \Sigma_{9,1} & \Sigma_{9,2} & \cdots & \Sigma_{9,9} & & & & & \\ & & & & \Sigma_{10,10} & & & & \\ & & 0 & & & \ddots & & & \\ & & & & & & \Sigma_{15,15} & & \end{bmatrix}$$

That is, β contains the unique elements of the upper 9 x 9 section and the diagonal elements of row 10 to 15. For a change, Δx, from x^*, the IFAP program generates $\hat{\beta}(x^* + \Delta x | x^*)$ the approximation to $\hat{\beta}(x^* + \Delta x)$. Then, for each Δx there are ML estimates $\hat{\beta}$ corresponding to $(\hat{\mu}, \hat{\Sigma})$. By comparing $\hat{\beta}$ and $\hat{\beta}$, the accuracy of the IFAP approximation can be studied. Since IFAP is a first-order estimator, $\hat{\beta}(x^* + \Delta x | x^*)$ and the ML estimate $\hat{\beta}(x^* + \Delta x)$ are expected to be different. However, $\hat{\beta}$ will approach $\hat{\beta}$ as Δx approaches zero.

II. Accuracy Demonstration

There are six cases in this section: in the first case, Δx has only one nonzero element in x; in the next three cases, the Δx represents a certain percentage change of all elements in one of the x's with respect to the baseline x_i^*; for the fifth case, Δx represents a change in one positive standard deviation for all elements of one x; the last case assumes one of the samples is abnormal and that $\hat{\beta}^* = \hat{\beta}(x^* + \Delta x)$, and then the IFAP approximation, $\hat{\beta}(x^* | x^* + \Delta x)$, is compared to the MLE, $\hat{\beta}(x^*)$.

CASE 1: The first element of x_1 was changed by 100% of its nominal value (x_1^*), i.e.,

$$\Delta x = \text{vec}\,(\Delta x_1, 0, \ldots, 0)$$

where vec denotes the vector form of a matrix, the zeroes to the right indicate that all the elements in that column are zeroes, and $\Delta x_1 = (x_{1_1}^*, 0, \ldots, 0)^T$.

Since only the first element of the first x was changed, the parameters μ_1 and $\Sigma_{1,1}$ were most affected. The $(\mu_1^*, \Sigma_{1,1}^*)$ values are (2.21, 223); the IFAP approximates are (2.57, 230); the MLEs are (2.59, 228). The normalize (defined below) differences between the IFAP approximation and nominal values are (.11, .09); the differences between the MLE and nominal values are (.12, .08). The normalization factors were the square roots of the appropriate diagonal element of the Fisher covariance matrix evaluated at the true μ and Σ values used in simulation. The changes in the other parameters are small, not exceeding .03 times their standard deviations.

CASE 2: The elements of x_{12} were changed by 50% of their nominal (x_{12}^*) values, i.e.,

$$\Delta x = \text{vec}\,[\ldots 0, .5x_{12}^*, 0, \ldots]$$

Table 1
Comparison of the IFAP and ML Solutions
50% changes for All Elements of x_{12}

STATE	BASE MEAN	MEAN ESTIMATES		NORMALIZED DELTA		BASE SIGMA	COVARIANCE ESTIMATES		NORMALIZED DELTA	
		ML	IFAP	ML	IFAP		ML	IFAP	ML	IFAP
1	2.21	1.84	1.87	-0.12	-0.11	223.	243.	241.	0.26	0.23
2	-7.12	-7.40	-7.32	-0.09	-0.06	178.	183.	184.	0.08	0.08
3	2.93	2.81	2.77	-0.04	-0.05	221.	225.	224.	0.06	0.05
4	-4.72	-4.96	-4.75	-0.08	-0.01	111.	112.	112.	0.02	0.02
5	-1.82	-2.52	-2.22	-0.22	-0.13	211.	241.	233.	0.39	0.28
6	3.18	3.25	3.42	0.02	0.08	389.	390.	390.	0.02	0.01
7	-3.19	-3.78	-3.84	-0.18	-0.20	290.	326.	315.	0.46	0.36
8	0.69	0.71	0.55	-0.06	-0.11	300.	304.	302.	0.0	0.04
9	-4.82	-5.13	-4.89	-0.10	-0.02	203.	201.	199.	-0.03	-0.05
10	1.01	0.87	0.77	-0.04	-0.09	267.	274.	274.	0.08	0.09
11	-3.44	-2.70	-2.76	0.23	0.21	179.	244.	232.	0.84	0.69
12	2.89	2.58	2.76	-0.10	-0.04	328	338	331.	0.11	0.07
13	-3.84	-3.92	-3.85	-0.02	-0.00	165.	16..	165.	0.00	.00
14	-2.71	-1.98	-2.03	0.23	0.21	460.	532.	515.	0.90	0.70
15	-3.63	-4.46	-4.21	-0.26	-0.18	107.	166.	152.	0.76	0.57

where 0 represents all elements in the column are zeros. The IFAP and ML estimates are given in Table 1. The BASE MEAN and BASE SIGMA in the table represent the baseline parameter estimates, $\hat{\mu}^*$ and $\hat{\Sigma}^*$. The updated ML and IFAP estimates for the means and variances represent the values for $\hat{\beta}_i(x^* + \Delta x)$ and $\hat{\hat{\beta}}_i(x^* + \Delta x|x^*)$. The DELTAS, $\hat{\beta}_i - \beta_i^*$ and $\hat{\hat{\beta}}_i - \beta_i^*$, are normalized by the appropriate standard deviation as described in CASE 1.

As shown in the table, the actual ML values and the IFAP estimates are fairly close. Also note that the sign of the deltas are the same in ML and IFAP in every parameter.

However, it is not immediate from the table how well IFAP works as a predictor of the relative sensitivities. That is, can IFAP accurately detect the parameter that is the most sensitive, the second most sensitive, etc.? Therefore, Figure 1 is a plot that compares the ranks of the estimates in terms of their sensitivities. Sometimes, the IFAP rank may not match the appropriate ML rank, even though the actual numerical differences between these two estimates are small. So, an error bar chart was added in Figure 1 underneath the rank plot.

The ranks of the parameters are assigned by their values of the normalized deltas from the largest negative value to the largest positive (i.e., rank "1" is assigned to the largest negative delta, rank "2" is assigned to the second largest negative, etc. until the largest positive delta is reached). Then, the IFAP vs ML ranks were plotted. If IFAP and ML ranks were perfectly matched, then the plotted points would stay on a 45° line; on the other hand, if the IFAP and ML ranks were completely unrelated, the plotted points would be scattered evenly throughout the area plotted.

The error bars show the absolute differences between the IFAP and ML estimates

$|\hat{\hat{\beta}}_i(x^* + \Delta x|x^*) - \hat{\beta}_i(x + \Delta x)|$, normalized by the appropriate standard deviation. If the numerical differences between the IFAP and ML estimates are small then the rank agreements are less important.

Note that, in terms of ranks, the greatest discrepancy between IFAP and ML occurs in the middle of the plot (see Figure 1). These parameters, however, also correspond to those that are least sensitive to Δx (and thus of least interest), and, as shown in the error bar chart, those for which the normalized errors between IFAP and ML are smallest. This greater discrepancy in ranks can be attributed to the interest variability associated with such small normalized errors.

CASE 3: All the elements of x_1 were changed by 50% of their nominal (x_1^*) values, i.e.,

$$\Delta x = \text{vec}\,[.5x_1^*,\ 0,\ \ldots,\ 0]$$

where 0 indicates that all elements in that column are zeros. This case is similar to CASE 2. The purpose of this case is to demonstrate that CASE 2 was fairly typical.

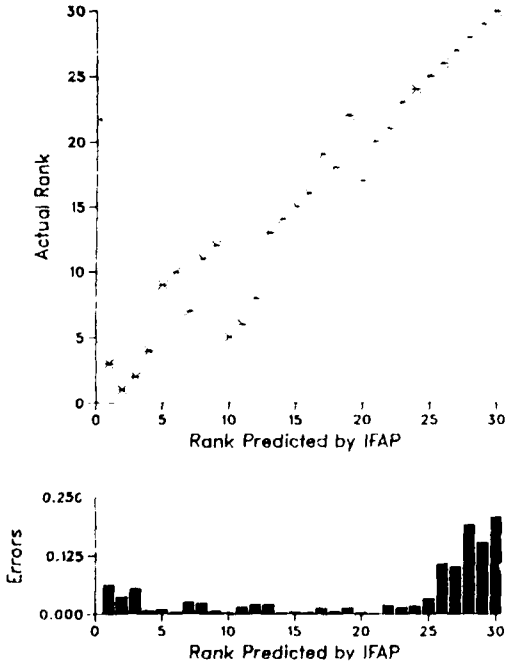


Figure 1: ML/IFAP Ranks and Normalized Errors 50% Changes for all Elements of x_{12}

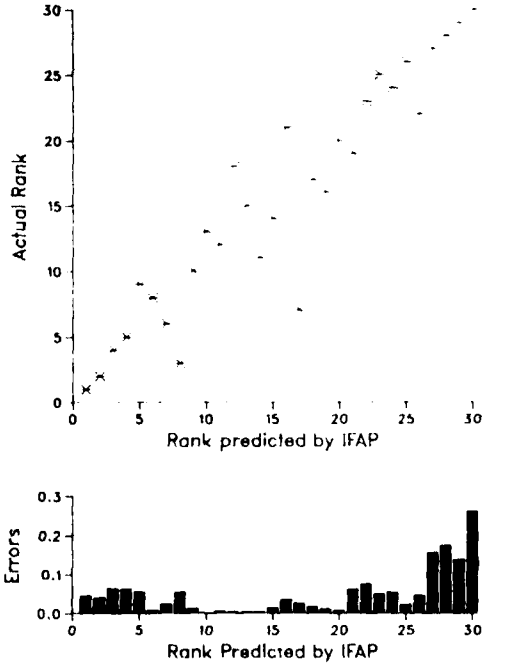


Figure 2: ML/IFAP Ranks and Normalized Errors 50% Changes for All Elements of x_1

Figure 2 shows the rank plot and the error bar chart of this case. The pattern in this figure is the same as in Figure 1. The lower left and upper right of the plot have many points lying on the 45° line while the error bars at center area are short and have the same magnitude errors as in CASE 2.

Since the plot and chart together convey the essential information for comparing IFAP and ML, a table such as Table 1 will be omitted in this case as well as in CASES 4 and 5.

CASE 4: All the elements of x_{12} were changed by 100% of their nominal (x^*_{12}) values, i.e.,

$$\Delta x = \text{vec}\,[\ldots, 0,\ x^*_{12},\ 0,\ \ldots]$$

In comparison with CASE 2, Δx is twice as large: this case also shows larger differences on the error bar chart, and a more scattered rank plot. Figure 3 shows that the largest error in the chart tripled in values, and there are 6 more points off the 45° line in the plot. However, it is apparent from the plot that there is still a strong tendency for the points in the rank plot to lie near the 45° line. The IFAP approximation has the same ranks as the MLE at the lower left and upper right corners of the plot. The points off the 45° line are concentrated at the center section and, as before, correspond to smaller normalized errors.

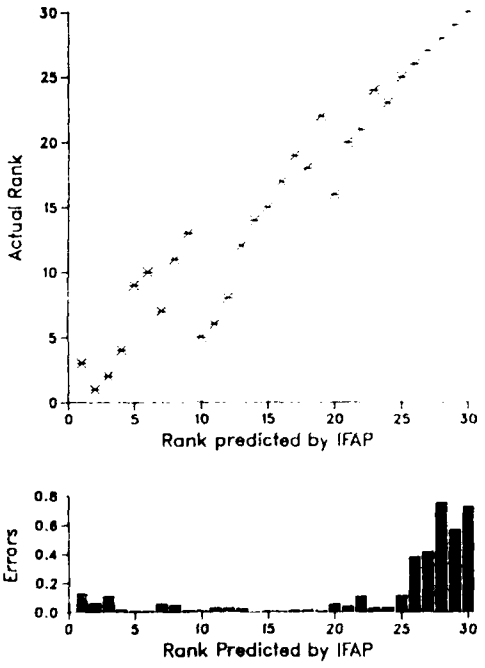


Figure 3: ML/IFAP Ranks and Normalized Errors 100% Changes for all Elements of x_{12}

CASE 5: All the elements of x_1 were changed by positive one standard deviation, i.e.,

$$\Delta x = \text{vec}[\Delta x_1,\ 0,\ \ldots,\ 0]$$

where

$$\Delta x_1 = [(\Sigma_{1,1} + P_{1_{1,1}})^{1/2},\ (\Sigma_{2,2} + P_{1_{2,2}})^{1/2},\ \ldots,\ (\Sigma_{p,p} + P_{1_{p,p}})^{1/2}]^T.$$

Some of the changes in the previous cases may be small in comparison to the standard deviations since 100% of a small value is still small. The changes of the elements in this case have the same ratio to the standard deviations; therefore, the rank plot at the top of Figure 4 is expected to be more evenly spread out than the previous plots. From this spread, the points in the plot still tend to stay along the 45° line and several of the most sensitive parameters have been assigned at the same ranks in both IFAP and ML. Referring to the error bar chart at the bottom of Figure 4, the points that tend to be off the 45° line in the plot have smaller errors, less than one-tenth of a standard deviation. Therefore, the IFAP approximation and MLE are closely matched.

CASE 6: Assume that all the x's have the same x^*_i's as in the previous cases (1-5), except x^*_3 is replaced by $2x^*_3$. Then all elements of $x_3 = 2x^*_3$ are changed back to original values, i.e.,

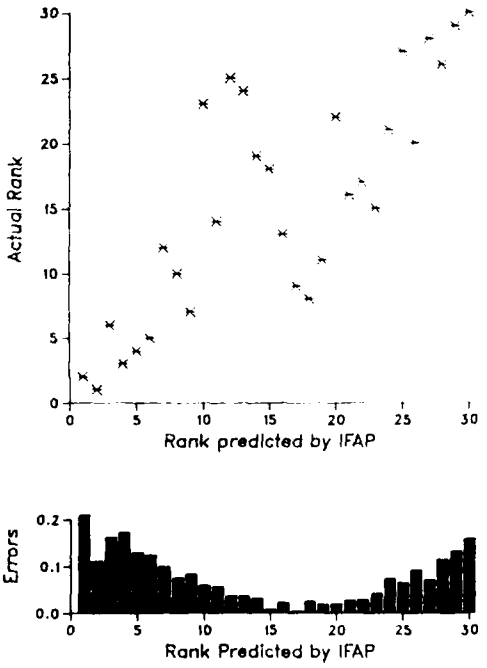


Figure 4: ML/IFAP Ranks and Normalized Errors Positive One Standard Deviation Changes for all Elements of x_1

$$\Delta x = \text{vec}[\,0,\ 0,\ -x_3^*,\ 0,\ \ldots]$$

This case was designed to show how IFAP would do when one of the x was an outlier. The comparison of the IFAP approximation and MLE is shown in Table 2.

Table 2
Comparison of the IFAP and ML Solutions
100% changes for All Elements of x_3

STATE	BASE MEAN	MEAN ESTIMATES		NORMALIZED DELTA		BASE SIGMA	COVARIANCE ESTIMATES		NORMALIZED DELTA	
		ML	IFAP	ML	IFAP		ML	IFAP	ML	IFAP
1	1.43	2.21	1.71	0.24	0.09	246.	223.	216.	-0.28	-0.38
2	-7.47	-7.12	-7.12	0.11	0.11	181.	178.	172.	-0.05	-0.12
3	4.43	2.93	3.37	-0.48	-0.34	338.	221.	176.	-1.51	-2.10
4	-5.20	-4.72	-4.84	0.15	0.11	116.	111.	103.	-0.07	-0.17
5	-2.26	-1.82	-1.91	0.14	0.11	224.	211.	201.	-0.17	-0.30
6	4.47	3.18	3.08	-0.41	-0.44	500.	389.	356.	-1.44	-1.86
7	-3.79	-3.19	-3.37	0.19	0.13	307.	290.	279.	-0.22	-0.35
8	1.92	0.89	0.95	-0.33	-0.31	360.	300.	284.	-0.77	-0.98
9	-4.11	-4.82	-4.89	-0.22	-0.25	246.	203.	194.	-0.55	-0.66
10	2.70	1.01	1.03	-0.54	-0.53	470.	267.	202.	-2.61	-3.46
11	-3.54	-3.44	-3.33	0.03	0.07	178.	179.	179.	0.00	0.01
12	2.58	2.89	3.13	0.10	0.17	331.	328.	328.	-0.04	-0.03
13	-4.33	-3.84	-4.08	0.15	0.08	169.	165.	156.	-0.05	-0.16
14	-3.48	-2.71	-2.78	0.24	0.22	487.	460.	451.	-0.34	-0.45
15	-4.14	-3.63	-3.76	0.16	0.12	120.	107.	101.	-0.16	-0.25

The baseline value in Table 2 is $\hat{\beta}^* = \hat{\beta}(x^* - \Delta x)$. The MLE have the same values as the previous baseline value $\hat{\beta}(x^*)$, and the IFAP approximation is $\hat{\beta}(x^* | x^* - \Delta x)$. The normalized deltas are the differences between the IFAP or ML estimates and $\hat{\beta}^*$; they are then divided by their standard deviations as described in CASE 1.

Table 2 shows that the IFAP and ML estimates are near one another even though $\hat{\beta}(x^* - \Delta x)$ is far from $\hat{\beta}(x^*)$, and that the (normalized) deltas for the IFAP and ML have the same signs. Thus, IFAP performs well for this outlier-type case, too.

III. The IFAP Results/Interpretation

There would be several ways to apply IFAP in actual data processing, e.g., approximating the estimates and studying the sensitivities. This section presents two tables that give some insight into such applications of IFAP. Recall that the IFAP program uses the same data as the ML estimator.

Most sensitivity studies require a large number of runs. Therefore, a study was made to investigate the efficiency of the IFAP program. The study, including 51 samples, 59 states and 154 parameters, shows that the IFAP CPU time was less than 1/25 of that required to generate an MLE by DL/scoring. For a case size like this large, an IFAP run took about 10 CPU seconds on IBM 3083. In other cases, of course, the CPU times may vary according to the number of x's, states, and parameters.

The two examples shown in this section are displayed in Tables 3 and 4. Each example was generated from nine similar IFAP runs. Every run was generated from the same baseline

Table 3

Relative Changes in β Parameters to Their Standard Deviations
for 100 Percent Changes in x (First 9 of 25 Samples)

SAMPLE(right) PARAMETER(down)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
μ_1	0.04	-0.06	-0.07	0.14	0.11	-0.01	0.50	0.26	0.43
μ_2	-0.36	-0.56	-0.12	0.28	-0.03	0.29	-0.09	-0.12	0.07
μ_3	0.53	0.22	0.31	-0.15	0.13	-0.04	0.23	0.07	-0.47
μ_4	-0.25	0.22	-0.09	0.15	-0.42	0.20	-0.24	-0.07	0.26
μ_5	-0.03	-0.02	-0.11	0.13	-0.21	0.09	0.40	0.16	-0.54
μ_6	0.33	-0.73	0.43	0.01	0.26	0.00	-0.00	-0.24	0.42
μ_7	0.07	-0.55	-0.11	-0.39	-0.17	-0.06	-0.25	0.13	0.25
μ_8	-0.32	0.12	0.32	-0.19	-0.07	-0.04	-0.31	0.55	-0.16
μ_9	0.09	-0.03	0.24	0.21	-0.35	-0.32	-0.17	0.15	0.22
μ_{10}	0.15	-0.42	0.53	0.20	-0.45	-0.02	-0.22	-0.04	0.25
μ_{11}	-0.27	-0.34	-0.08	-0.07	0.12	-0.20	0.08	0.32	0.09
μ_{12}	0.50	0.24	-0.19	-0.01	0.59	-0.42	-0.01	-0.07	0.26
μ_{13}	0.06	[illegible]	[illegible]	-0.17	[illegible]	-0.20	-0.42	-0.24	-0.00
μ_{14}	-0.11	-0.21	-0.24	-0.03	-0.19	-0.33	0.40	-0.26	-0.64
μ_{15}	-0.23	-0.26	-0.11	0.35	-0.15	-0.15	-0.17	0.11	0.04
$\|\mu\|_1$	3.34	4.53	[illegible]	2.50	3.60	2.37	3.49	2.78	4.12
$\Sigma_{1,1}$	0.05	0.04	0.23	0.24	0.23	0.02	1.17	0.54	0.49
$\Sigma_{2,2}$	0.98	1.19	0.05	0.65	0.04	0.83	0.03	0.02	0.21
$\Sigma_{3,3}$	1.42	0.36	0.90	0.25	-0.13	0.00	0.14	0.01	0.77
$\Sigma_{4,4}$	0.17	0.07	-0.05	0.37	1.52	0.21	0.43	-0.02	0.15
$\Sigma_{5,5}$	-0.12	0.08	0.09	-0.01	0.10	0.06	1.12	0.24	1.75
$\Sigma_{6,6}$	1.21	2.43	1.02	0.26	-0.03	-0.06	0.01	0.37	0.59
$\Sigma_{7,7}$	0.27	0.66	0.17	0.78	0.10	3.01	0.12	0.14	0.34
$\Sigma_{8,8}$	0.36	-0.08	0.57	0.16	0.14	0.09	0.67	1.51	0.24
$\Sigma_{9,9}$	0.00	-0.01	0.46	0.34	0.43	0.57	0.18	0.30	0.10
$\Sigma_{10,10}$	0.12	0.65	1.80	0.53	0.70	-0.01	0.25	0.05	0.39
$\Sigma_{11,11}$	0.18	0.44	-0.06	0.15	-0.00	0.17	-0.10	0.39	0.09
$\Sigma_{12,12}$	1.94	0.20	0.08	0.25	1.46	1.17	-0.01	0.06	0.46
$\Sigma_{13,13}$	0.07	1.56	-0.05	0.02	0.17	0.04	0.95	0.20	0.03
$\Sigma_{14,14}$	-0.14	0.30	0.22	-0.12	-0.06	0.52	0.57	0.56	2.26
$\Sigma_{15,15}$	0.04	0.18	0.03	0.78	0.12	0.03	0.16	-0.05	0.05
$\|\Sigma\|_1$	7.07	8.05	5.77	4.89	5.23	3.76	5.90	4.47	7.93

Table 4

Relative Changes in β Parameters to Their Standard Deviations
for 100 Percent Changes in x (First 9 of 25 Samples)

SAMPLE(right) PARAMETER(down)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
μ_1	-0.03	0.08	0.09	-0.13	-0.06	0.02	0.49	-0.24	-0.42
μ_2	0.34	0.56	0.11	-0.29	0.03	-0.29	0.08	0.12	-0.07
μ_3	-0.55	-0.21	-0.34	0.14	-0.11	0.03	-0.24	-0.07	0.45
μ_4	0.22	-0.22	0.11	-0.16	0.41	-0.21	0.24	0.08	-0.26
μ_5	0.00	0.01	0.11	-0.13	0.17	-0.08	-0.40	-0.16	0.55
μ_6	-0.36	0.74	-0.44	-0.02	-0.25	-0.01	0.01	0.23	-0.42
μ_7	-0.07	0.55	0.11	0.38	0.17	0.07	0.25	-0.13	-0.25
μ_8	0.27	-0.12	-0.31	0.19	0.08	0.04	0.31	-0.54	0.16
μ_9	-0.09	0.01	-0.25	-0.21	0.32	0.32	0.17	-0.15	-0.22
μ_{10}	-0.16	0.43	-0.53	-0.20	0.48	0.00	0.22	0.04	-0.24
μ_{11}	0.26	0.34	0.07	0.07	-0.14	0.21	-0.07	-0.31	-0.10
μ_{12}	-0.50	-0.24	0.17	-0.00	-0.58	0.40	0.01	0.08	-0.26
μ_{13}	-0.09	0.56	0.08	0.17	0.34	0.19	0.42	0.25	-0.00
μ_{14}	0.11	0.24	0.22	0.04	0.20	0.32	-0.40	0.26	0.64
μ_{15}	0.20	0.24	0.12	-0.37	0.15	0.15	0.16	-0.10	-0.05
$\|\mu\|_1$	3.28	4.57	3.06	2.51	3.50	2.35	3.46	2.75	4.09
$\Sigma_{1,1}$	0.07	0.02	0.38	-0.25	-0.21	-0.02	-1.18	-0.54	-0.52
$\Sigma_{2,2}$	-0.93	-1.17	-0.12	-0.68	-0.01	-0.84	-0.02	-0.02	-0.22
$\Sigma_{3,3}$	-1.31	-0.15	-2.10	-0.25	0.11	0.05	-0.13	-0.00	-0.79
$\Sigma_{4,4}$	-0.15	-0.66	-0.17	-0.39	-1.42	-0.24	-0.41	-0.02	-0.17
$\Sigma_{5,5}$	0.15	-0.10	-0.30	0.01	-0.22	-0.05	-1.14	-0.24	-1.72
$\Sigma_{6,6}$	-1.16	-2.53	-1.86	-0.27	0.01	0.04	-0.05	-0.41	-0.54
$\Sigma_{7,7}$	-0.24	-0.65	-0.35	-0.74	-0.08	-0.00	-0.12	-0.16	-0.36
$\Sigma_{8,8}$	-0.38	0.06	-0.98	-0.20	-0.19	-0.14	-0.67	-1.46	-0.26
$\Sigma_{9,9}$	0.01	-0.01	-0.66	-0.35	-0.45	-0.53	-0.18	-0.28	-0.08
$\Sigma_{10,10}$	0.08	-0.22	-3.46	-0.50	0.79	0.06	-0.31	-0.08	-0.30
$\Sigma_{11,11}$	-0.19	-0.43	0.01	-0.16	-0.02	-0.17	0.11	-0.39	-0.10
$\Sigma_{12,12}$	-1.94	-0.20	-0.03	-0.29	-1.51	-1.13	0.02	-0.06	-0.48
$\Sigma_{13,13}$	-0.09	-1.53	-0.16	-0.03	-0.16	-0.03	-0.93	-0.20	-0.04
$\Sigma_{14,14}$	0.07	-0.34	-0.45	0.11	0.05	-0.50	-0.58	-0.53	-2.20
$\Sigma_{15,15}$	0.04	-0.16	-0.25	-0.84	-0.11	-0.05	-0.15	0.05	0.06
$\|\Sigma\|_1$	6.82	8.12	11.28	5.08	5.24	3.86	4.99	4.44	7.89

$\hat{\beta}^*$, x^*, and P_i's. Each column corresponds to the one sample that was changed by 100% in generating $\hat{\beta}$; the other samples remain at their x^* values. Table 3 was generated using the x_i^*'s and P_i's as in Section 2. Table 4 was generated with a modified x_3^*, the elements of which were doubled in comparison with the values of Table 3. This will illustrate how IFAP performs in the presence of an outlier (sample 3).

Tables 3 and 4 show the normalized deltas, i.e., the difference between $\hat{\beta}$ and $\hat{\beta}$, normalized by the Fisher-based standard deviations as described in CASE 1. The columns headed by x_1, x_2, ..., x_9 correspond to samples, 1, 2, ..., 9. The values for $\|\mu\|$ and $\|\Sigma\|$ at the bottom of each column denote the sum of the absolute values of the entries in that column.

Table 3 was generated using $\hat{\beta}^* = \hat{\beta}(x^*)$ and $x_i^* \sim N(0, P_i + \Sigma)$ for i = 1, 2, ..., 25. Notice that the $\|\Sigma\|$'s are about twice as large as the $\|\mu\|$'s within all samples. The differences in $\|\Sigma\|$ and $\|\mu\|$ may be largely attributed to the relationship between Δx and $\Delta\mu$ and Δx and $\Delta\Sigma$. Recall that the equation $\frac{\partial L}{\partial \mu} = 0$ implies that Δx and $\Delta\mu$ have a linear relationship, while the equation $\frac{\partial L}{\partial \Sigma} = 0$ implies that Δx and $\Delta\Sigma$ have approximately a second-order relationship.

Table 4 was generated using $\hat{\beta}^* = \hat{\beta}(x^* + \Delta x_3)$, $x_i^* \sim N(0, P_i + \Sigma)$ for i = 1, 2, ..., 25 and $\Delta x_3 = \text{vec}[0, 0, x_3^*, 0, \ldots]$. The purpose of this example is to show the outcome of IFAP when there is an outlier, Sample 3, in the system. In real data Analysis, outliers may dominate the result and lead to erroneous conclusions. Although IFAP is not designed specifically for isolating outliers, the unusually large value of $\|\Sigma\|$ and the ratio of $\|\Sigma\|$ to $\|\mu\|$ for sample 3 reflect the fact that this sample is an outlier.

4. CONCLUSION

This study demonstrates that IFAP can be an effective and efficient tool for studying the impact of anomalous data on the ML estimates of means and variances.

In Section 3.II it was shown that the current first-order implementation provides an accurate approximation to the changes in parameter estimates resulting from changes in the data of a selected x. It was found that if the parameters were ranked in order of their sensitivities to these changes in data, the ranks of the parameters as given by IFAP were close to the ranks as given by recalculating MLEs. This was especially true among those parameters that were most sensitive to data changes, which, of course, would correspond to the parameters of most interest.

Section 3.III demonstrates how IFAP might apply in actual data Analysis. In particular, two tables were presented that illustrate how IFAP can be used to show at a glance how sensitive various parameter estimates are to changes in the data of one x. As an aside, it was found that we were able to detect an outlier x by its abnormal impact on the estimate of the variance terms; we have not yet developed a rigorous theoretical basis for this observed phenomenon. We found that it was approximately 25 times more efficient (in terms of CPU time) to calculate updated IFAP estimates than to calculate updated MLEs in a larger size problem. This may be the difference between feasibility and infeasibility in a large-scale data sensitivity study.

ACKNOWLEDGEMENT

This work was supported by U.S. Navy contract N00024-85-C-5301 task PM.

REFERENCES

Iman, Ronald L. and Conover, W. J. [1980]. "Small Sample Sensitivity Analysis Techniques for Computer Models, with an Applications to Risk Assessment," Communication in Statistics - Theory and Methods, A9 (17), 1749-1842.

Kalaba, Robert and Tesfatsion, Leigh [1986]. "Nonlocal Sensitivity Analysis Automatic Derivative Evaluation, and Sequential Nonlinear Estimation," Computational Statistics and Data Analysis 4, 79-91.

Spall, J. C. [1985]. "An Implicit Function Based Procedure for Analyzing Maximum Likelihood Estimates from Nonidentically Distributed Data," Communication in Statistics- Theory and Methods, Vol. 14, 1719-1730.

Spall, J. C. [1986]. "An Approximation for Analyzing a Broad Class of Implicitly and Explicitly Defined Estimates," Communications in Statistics - Theory and Methods, 15 (12), 3747-3762.

BOOTSTRAP PROCEDURES IN RANDOM EFFECT MODELS FOR COMPARING RESPONSE RATES IN MULTI-CENTER CLINICAL TRIALS

Michael F. Miller
Hoechst-Roussel Pharmaceuticals Inc.
Somerville, New Jersey 08876

1. INTRODUCTION

Clinical trial designs for comparing an experimental treatment with an appropriate control commonly use several investigators located at a variety of medical centers, all operating from the same protocol. This paper is concerned with treatment versus control comparisons based on a dichotomous response. A specified event, termed a response in this paper, is observed to have occurred or not occurred for each subject in the trial. The context for a statistical comparison of treatment and control is a stochastic model assuming treatment and control probabilities of response at each center.

DerSimonian and Laird (1986) observe that the control and treatment response probabilities will likely vary from center to center, or vary from study to study in a meta-analysis of similar clinical trials. They propose a random effects model assuming that a center's control and treatment response probabilities are themselves random variables with a distribution dependent on the population of centers under study.

Let ‹P,Q› be the control and treatment response probabilities at a given center. The pair ‹P,Q› are themselves random variables with:

(1.1) joint distribution of ‹P,Q› = g(p,q), ‹p,q› varying in the unit square, or in a subset of the unit square.

Following the selection of ‹P,Q›, independent samples of n control and m treatment subjects are observed. If X, Y are the observed frequencies of the control and treatment responses, then X and Y are assumed to have independent binomial p.d.f.'s conditioned on the assumed values P=p, Q=q.

If k centers are planned for a multi-center trial, then the unobserved response probabilities ‹Pj,Qj›, j=1,k are assumed i.i.d from g, while Xj,Yj are the observed control and treatment response frequencies from nj,mj subjects at center j.

This paper explores the use of the bootstrap method (Efron, 1982) to compute significance levels and confidence intervals for statistical inference problems generated by the above model. Parameters are defined in terms of the random effects density g, and estimates of these parameters are generated from estimates of g based on the observed response frequencies. An important special case is studied first: the proportional odds assumption, where the treatment to control odds ratios are assumed to be homogeneous across centers. Nonparametric versions of the bootstrap are then explored for the more general random effects model when the proportional odds assumption cannot be used.

Two examples will be given illustrating the use of these methods. One example involves a multi-center trial, the other example is a meta-analysis of several trials discussed in DerSimonian and Laird's paper. For this meta-analysis the sampling unit for the random effects model is a particular study rather than a study site. The models and methods used here are formally the same for a meta-analysis as they are for the multi-center trial although the interpretation of the results can be different.

2. PROPORTIONAL ODDS MODELS

Suppose, for the random pair ‹P,Q›, the following can be assumed:

(2.1) Q/(1-Q) = r*P/(1-P), r a fixed positive constant.

Here the odds for occurrence in the treatment group is a constant multiple of the control odds for occurrence, and r is the constant odds ratio, treatment to control. Under this assumption the random pair ‹P,Q› must vary within a one dimensional subset (curve) of the unit square.

This important special case has been studied extensively in connection with the Mantel-Haenszel test. Note that the hypothesis of r=1 in the proportional odds model implies Pj=Qj for every center j=1,2,..k. This is the usual "no treatment effect" null hypothesis for the Mantel-Haenszel test. Wittes and Wallenstein (1987) discuss approximations to the power of this statistic and give an excellent reading list on this subject.

Under the proportional odds assumption inferences about the fixed odds ratio r (or log odds ratio lr = ln(r)) do not require a random effects model even though the response probabilities can vary considerably from center to center. The conditional likelihood function given the assumed values P1=p1, P2=p2,....Pk=pk, and

$$qj = r*pj/(1-pj+r*pj)$$

can be expressed entirely in terms of the control response rates, pj j=1,..k, and the common odds ratio r. The maximum likelihood estimates of r and p1, p2,...pk cannot be found in closed form, but an elementary numerical iteration can be used to calculate the estimates and their standard errors. Bootstrapping the sampling distribution of the MLE of r, and the Mantel-Haenszel estimate of r given in Fleiss, 1981, suggests their sampling distributions are very similar for examples involving moderate within site sample sizes.

Computation of the MLE's permits the use of the likelihood ratio test for assessing the goodness of fit of the proportional odds assumption. In practice this test should be made at a level higher than p=.05 so that the greater sensitivity available under homogeneous odds ratios is not so easily assumed. Examples of these methods are given in section 5. The next section discusses inferences when the odds ratios cannot be assumed to be homogeneous. Because sample logits tend to be more normally distributed than odds ratios, log odds ratios will be used from now on.

3. A NONPARAMETRIC RANDOM EFFECTS FORMULATION

Consider again the random response probabilities ‹Pj,Qj› j=1,...k as a random sample from the joint p.d.f. g defined in (1.1). The null hypothesis of no treatment effect proposed here is given by:

$$(3.1) \quad g(p,q) = g(q,p).$$

Symmetry of the joint p.d.f. about p=q conveys the essential meaning of no treatment effect. Note that any real valued transformation having the property T(p,q)=-T(q,p) yields a distribution symmetric about zero for T(P,Q) under (3.1) and this in fact characterizes (3.1). In particular, T(p,q)=ln(q/(1-q)) - ln(p/(1-p)) = log odds ratio satisfies this property. Estimates of g under (3.1) are proposed in section 4. Estimates of the joint p.d.f. g in general are formulated now in terms of the estimated control and treatment response rates:

$$PHj = (Xj+.5)/(nj+1)$$
$$QHj = (Yj+.5)/(mj+1), \quad j=1,....k.$$

Three nonparametric estimates of g are considered in this paper. Each of these estimated joint p.d.f.s assigns all of its mass to the subset of observed response rate pairs:

$$SPRT = \{ \langle PHj,QHj \rangle : \ j=1,..k \}.$$

(3.2) GH1(p,q) = 1/k, ‹p,q› in SPRT,

(3.3) GH2(p,q) = c*nj*mj/(nj+mj), ‹p,q› = ‹PHj,QHj›, where c is the constant making GH2 sum to 1.0 over SPRT,

(3.4) GH3(p,q) = maximum likelihood estimate of g among p.d.f.s assigning all mass within SPRT.

Estimates (3.2), (3.3) are computationally simple, and are consistent when k and the min{n1,...nk,m1,....mk} diverge to infinity. The details of this are not relevant here because while the nj's and mj's are often large, k=number of centers is usually small.

The maximum likelihood estimate specified by (3.4) can be obtained from the EM algorithm (Dempster, Laird, Rubin, 1977), but a closed form solution is available also. The likelihood function that must be maximized has the form:

$$mj(g) = \prod_j \sum_w g_w \, b_{wj},$$

where

$$g_w = g(pw,qw),$$

$$b_{wj} = pw^{xj}(1-pw)^{nj-xj} \times qw^{yj}(1-qw)^{mj-yj},$$

‹pw,qw› in SPRT.

The details of this will be given in another paper.

Once an estimate of g is obtained, namely GH using either (3.2), (3.3), or (3.4), then estimates of the log odds ratio can be formulated. In this section the log odds ratio is not constant. Let

$$(3.5)\quad lr(g) = E[\ln(Q/(1-Q)) - \ln(P/(1-P)) : g],$$

where the expected value is taken with respect to g, and g is such that lr(g) is finite. Define

$$(3.6)\quad LRH = lr(GH)$$

as the estimate of the mean log odds ratio based on GH. Note that LRH is just a weighted average of the empirical log odds ratios, the weights provided by GH. The next section discusses how the sampling distribution of LRH can be approximated for forming confidence intervals and computing significance levels.

4. APPLICATION OF THE BOOTSTRAP

The role of the bootstrap here is to provide an approximation to the sampling distribution of LRH, where both this sampling distribution and LRH are determined by GH. The bootstrap distribution can be an imperfect substitute for the unknown sampling distribution of LRH determined by g, the true underlying random effects p.d.f. With small k there is no guarantee that the measure defined by GH is anything like the measure defined by g. There is also the issue of whether a g exists, whether sites chosen for a clinical trial are representative of any real population, with some p.d.f. g. Note that these problems of small k and a population of sites disappear under proportional odds because the odds ratio MLE was driven entirely by the conditional likelihood given the assumed values of the site response probabilities. However, when proportional odds cannot be assumed, using GH as a working model in a random effects setting can be more credible than the usual Mantel-Haenszel tests even with the small k and the artifactual nature of GH.

In the realm of heterogeneous odds ratios, the conclusions derived from a data analysis may depend heavily on the selection of the method for estimating g. The bootstrap readily provides answers to the inference problems within any "computable" empirical model selected for analysis, and therefore provides the ability to assess how the general conclusions regarding treatment effect depend on the selection of this model, as will be illustrated by examples in section 5. These examples will also illustrate the price in precision that must be paid in moving away from a proportional odds assumption.

The general algorithim for the bootstrap used here is as follows for the random effect situation. The percentile - t method of generating interval estimates for lr(g) will be used in order to take advantage of any reduction in coverage probability error (Beran, 1987).

(4.1) Obtain an estimate, GH, as in (3.2), (3.3), or (3.4). Compute LRH and SEH, an asymptotic approximation to the standard error of LRH.

$$SEH = SQRT\ \{\Sigma_j\ GH_j^{2*}(vh_j + \sigma^2(GH))\},$$

where

$$vhj = 1/(nj*PHj*(1-PHj)) + 1/(mj*QHj*(1-QHj)),$$

$$\sigma^2(g) = Variance\ (\ln(Q/(1-Q)) - \ln(P/(1-P)) : g).$$

(4.2) Sample i.i.d. k pairs <PBj,QBj> from GH.

(4.3) For each j, sample nj,mj binomial trials using response probabilities <PBj,QBj> and note XBj,YBj, the response frequencies, j=1,2,...k.

(4.4) Using the data obtained from (4.3) compute the estimate GHB in the same way GH was computed from the original data. Then compute LRHB and its approximate standard error SEHB from GHB. Also compute ZB=(LRH-LRHB)/SEHB, the studentized transformation.

Repeat (4.2, 3, 4) NB times (I used NB=600) to obtain empirically the sampling distribution of LRH when using GH as the random effects p.d.f. The empirical distribution of the ZB's is used to form a percentile-t interval estimate of lr(g). If Z(2.5) and Z(97.5) are the 2.5th and 97.5th percentiles of the ZB's then

$$(4.5)\quad \{LRH + Z(2.5)*SEH,\ LRH + Z(97.5)*SEH\}$$

is an approximate 95% confidence interval for lr(g).

To obtain a test of the null hypothesis given in (3.1), namely $g(p,q)=g(q,p)$, then the following estimation procedure is used for g assuming the null hypothesis is true. Let GH again be an estimator of g with support SPRT.

(4.6) First translate the points in SPRT so that the center of mass falls on the line p=q.

$$PHTj = PHj + (mean[QH:GH] - mean[PH:GH])/2$$

$$QHTj = QHj + (mean[PH:GH] - mean[QH:GH])/2$$

(4.7) Now define the estimated null distribution GHO, $GHO(p,q) = (1/2)*GH(p,q)$, for $\langle p,q\rangle=\langle PHTj, QHTj\rangle$ or $\langle QHTj,PHTj\rangle$ $j=1,2,\ldots k$. $GHO(p,q)=0$ elsewhere.

Note that GHO is just the original GH equally divided among the translated points and their reflections through p=q, and by construction satisfies the null hypothesis. A reflection of the points in SPRT without a translation would create too wide a dispersion for GHO if most of the points in SPRT were far from the line p=q. This would result in an unnecessarily heavy tailed null distribution for LRH.

To obtain an empirical one tailed significance level for the estimate LRH repeat steps (4.2), (4.3), (4.4) NB times sampling from GHO instead of GH. Here the empirical distribution of the LRHB's will be symmetric about zero and can be used directly to compute the significance level. If large values of LRH are expected with a treatment effect, then the bootstrap significance level is:

(4.8) phb = (number of LRHB's exceeding LRH)/NB.

Again if k and the nj's, mj's are large then phb will be close to the actual attained significance level under the null hypothesis. With small k this method provides an internally consistent approximation within the context of sampling from GHO. Using several methods to obtain GH as discussed in section 3 it is possible to obtain several values of phb to see if the overall treatment effect conclusion is affected by the method of estimation. These procedures are illustrated in the next section.

5. TWO EXAMPLES

The first example involves a test drug for treating ulcers. The following data was obtained after two weeks of treatment.

STUDY SITE	PLACEBO NO. HEALED/N	%	DRUG NO. HEALED/N	%	LOG ODDS RATIO DRUG TO PLACEBO
1	11/33	33.3	24/37	64.9	1.31
2	2/25	8.0	5/26	19.2	1.01
3	5/28	17.9	7/23	30.4	0.70
4	4/16	25.0	3/14	21.4	-0.20
5	7/17	41.2	6/21	28.6	-0.56
6	3/23	13.0	4/24	16.7	0.29

This drug was clearly effective after 4 weeks of treatment. The question here is whether an efficacy claim is warranted after two weeks.

The likelihood ratio test for proportional odds yields a chi-square statistic = 6.09 with 5 degrees of freedom, clearly not significant at p=0.2. Assuming proportional odds, the likelihood ratio chi-square statistic testing the hypothesis that r=1 is 4.31 with 1 degree of freedom, significant at p=.05. The estimated common log odds ratio is .58 +- .28 with a 95% confidence interval of (0.03, 1.13). This analysis suggests a claim for efficacy relative to placebo can be made after two weeks of treatment.

What is disquieting about this conclusion is that the last three study sites did not yield overwhelming evidence for the drug. The lack of significance in the test for proportional odds may be due more to small sample sizes rather than homogeneity of odds ratios.

The three estimators given by (3.2), (3.3), and (3.4) were used in the context of a random effects model. The following results were obtained for the expected log odds ratio.

TWO WEEKS ON TREATMENT
MEAN LOG ODDS RATIO ESTIMATES

TYPE OF ESTIMATE		ESTIMATE ± SE	95% CONFIDENCE INTERVAL
Equal Wghts	(3.2)	0.39 ± 0.39	(-0.29, 1.09)
n*m/(n+m)	(3.3)	0.54 ± 0.39	(-0.14, 1.26)
MLE	(3.4)	0.29 ± 0.53	(-0.45, 0.94)

The standard errors were obtained via the asymptotic approximation, and the confidence intervals were obtained using the percentile-t bootstrap method given by (4.5). With the exception of the MLE, these asymptotic standard errors were in agreement with the corresponding bootstrap standard errors. The bootstrap standard error for the MLE was 0.38, somewhat lower than 0.53 given above.

Bootstrap significant levels were obtained using (4.8).

ESTIMATED P LEVEL

Equal Wgts	0.15
n*m/(n+m)	0.09
MLE	0.27

The confidence intervals and significance levels do not support a claim for efficacy because the estimators GH tend to emphasize the variability in the log odds ratios. Note that the Equal Weights estimator gives more weight to the negative studies than the second estimator which weights the sites according to a sample size factor. Note also that this second estimate is closer than all the others to the proportional odds estimate, because the sites are weighted in a manner similar to the Mantel-Haenszel method.

The random effects MLE (3,4) assigned weights according to the following proportions:

SITE	1	2	3	4	5	6
MLE	.167	.000	.071	.135	.125	.501

Most of the weight was pulled toward site 6, where the results for the drug were not spectacular. Why site 1 got one sixth of the weight and site 6 got 50% of the weight is a subject for another paper.

In summary, this first example seemed to satisfy the proportional odds assumption which, when applied, led to a conclusion of drug efficacy after two weeks on treatment. This conclusion relied heavily on proportional odds because all three random effect analyses yielded nonsignificant evidence for efficacy.

The second example, discussed by DerSimonian and Laird, is a meta-analysis of placebo controlled trials testing the effectiveness of cimetidine for healing ulcers (Winship, 1978). The following data was taken from this study.

STUDY	PLACEBO NO. HEALED/N	%	DRUG NO. HEALED/N	%	LOG ODDS RATIO DRUG TO PLACEBO
1	8/19	42.1	16/19	84.2	1.99
2	5/14	35.7	26/30	86.7	2.46
3	12/20	60.0	17/20	85.0	1.33
4	5/18	27.8	17/20	85.0	2.69
5	7/24	29.2	47/65	72.3	1.85
6	4/21	19.0	13/21	61.9	1.93
7	16/42	38.1	36/43	83.7	2.12
8	55/142	38.7	74/130	56.9	0.74

This example is interesting because the proportional odds assumption is rejected by the data, but this should not stand in the way of observing that cimetidine was significantly more effective than placebo. The chi-square test for proportional odds was 15.85 with 7 degrees of freedom, significant at the .05 level. Since a common log odds ratio is rejected by this data, estimates of a mean log odds ratio are given.

EFFECT OF CIMETIDINE
MEAN LOG ODDS RATIO ESTIMATES

TYPE OF ESTIMATE		ESTIMATE ± SE	95% CONFIDENCE INTERVAL
Equal Wghts	(3.2)	1.79 ± 0.30	(1.31, 2.32)
n*m/(n+m)	(3.3)	1.41 ± 0.37	(0.77, 2.13)
MLE	(3.4)	1.74 ± 0.39	(1.21, 2.08)

As evidenced by the estimates and the confidence intervals, the estimated mean log odds ratio is sufficiently far away from zero regardless of which method of estimation is used. All three bootstrap significance levels were less than .001.

The estimate (MLE) of the log odds ratio under the erroneous assumption of proportional odds is 1.41 +- 0.17. In the random effects model the conclusion of cimetidine efficacy is still apparent even after paying a substantial penalty in the standard error. Note that again the second estimator (weights prop. to n*m/(n+m)) gives a value similar to the MLE under proportional odds.

The maximum likelihood estimate (3.4) of the random effect distribution was:

STUDY	3	5	6	7	8
MLE	.062	.225	.036	.529	.148

Studies 1,2, and 4 received zero weight.

The SAS programs used in this paper are available by request from the author.

REFERENCES

Efron, B. (1982) The Jacknife, the Bootstrap, and Other Resampling Plans. SIAM, Philadelphia.

DerSimonian, R. and Laird, N. (1986) "Meta-analysis in clinical trials". Controlled Clinical Trials, 7: 177-188.

Wittes, J. and Wallenstein, S. (1987) "The power of the Mantel-Haenszel test". Journal of the American Statistical Association, 82, 400: 1104-1109.

Winship, D. (1978) "Cimetidine in the treatment of duodenal ulcer". Gastroenterology 74: 402-406.

Fleiss, J. (1981) Statistical Methods for Rates and Proportions. John Wiley & Sons, New York.

Dempster, A., Laird, N., Rubin, D. (1977) "Maximum likelihood from incomplete data via the EM algorithm". J. Royal Statistical Society B. 39: 1-38.

Beran, R. (1987) "Prepivoting to reduce level error of confidence sets". Biometrika, 74, 3: 457-468.

Bootstrapping the Mixed Regression Model with Reference to the Capital and Energy Complementarity Debate*

Baldev Raj, Wilfrid Laurier University

1. INTRODUCTION

The estimation of the partial Allen elasticity of substitution between energy and capital in the manufacturing process has been the subject of a number of studies. The results from these studies have not always been in agreement. For example, Berndt and Wood (1975) found that capital and energy were complements, while Griffin and Gregory (1976) and Pindyck (1979), found that capital and energy are substitutes. The implications of energy and capital complementarity is that ceteris paribus, higher priced energy will not only dampen its own demand, but also the demand for new investment in plants and equipment.

A number of avenues for reconciling these conflicting empirical results have been explored in the literature. For example, it has been suggested that the use of time series versus cross-section data lead to different results; studies that use time series data capture short-run factor relationships while studies that use cross-section data measure the long-run factor relationships. Others have argued that there is a need to disaggregate capital inputs into physical and working capital. The hypothesis is that while physical capital is complementary to energy, working capital is a substitute for energy. Others have stressed the need to exclude taxes from the capital working service price. Similarly, the need to use four inputs instead of three has been suggested. These and other arguments are reviewed by Solow (1987).[1]

In this paper we examine the sensitivity of the energy-capital complements issue by estimating the partial Allen elasticity of substitution between inputs i and j (σ_{ij}) under stochastic constraints[2] on the coefficients of the conditional input demand (CID) functions. The stochastic constraints are imposed corresponding to homogeneity and symmetry hypotheses; the estimates of σ_{ij}'s are obtained by using time series data covering the period from 1947-71. The data are obtained from Berndt and Wood (1975). A novelty of this paper is the use of the bootstrap (Efron, 1979) to estimate the standard error of the estimate of σ_{ij}. A case for using stochastic constraints instead of fixed (or exact) constraints has been made by many researchers including Tsurmi et al. (1986) and Ilmakunnas (1986). It can be argued that the use of exact constraints, which are a special case of stochastic constraints approach are both restrictive and unnecessary. Our examination builds on the papers by Freedman and Peters (1984) and Ilmakunnas (1986) who have used similar methods to those in this paper in a different but related context. We estimated σ_{ij}'s by the mixed estimation[3] (MR) method (Theil and Goldberger, 1961) to show that the estimates of σ_{ij} are sensitive to choice of a key parameteric value in the stochastic constraints. This parameter may be interpreted as a coefficient of stickiness towards homogeneity and symmetry hypotheses. Our results show that when the stickiness coefficient is assigned a value higher than those in the sample the estimate of σ_{KE} can be positive instead of negative. Further, its 75% confidence intervals $[\tilde{\sigma}_{KE} \pm t_{60}^{.75} \text{ SE } (\tilde{\sigma}_{KE})]$ fail to exclude a positive value for the $\tilde{\sigma}_{KE}$ either when the standard error of $\tilde{\sigma}_{KE}$ (SE) from the standard asymptotics or bootstrap (Efron, 1979) is used.[4] This result shows that energy-capital substitutability *cannot* be ruled out for this configuration of the stickiness coefficient which might be interpreted to reflect higher perceived or real uncertainty, asymmetric information or institutional stickiness faced by firms. The confidence intervals of σ_{KE} continue to include positive value of elaticity when fatter-tailed errors are considered. The fat-tailed errors are said to arise where sample data include unusual events such as oil price shock, oil emgargo, etc. (Taylor, 1983).

The paper is organized as follows: following this section we present the model and describe the MR estimation technique. In Section 3 we present the results and their discussion; this section also includes a brief review of the bootstrap idea. Final remarks conclude the paper.

2. THE MODEL AND MR ESTIMATION METHOD

The CID functions for the transdental logarithmic (Cristensen et al 1971) unit cost function are given by:

$$(1) \qquad S_i = \alpha_i + \sum_j \beta_{ij} \ln w_j + \epsilon_i$$

where S_i is the cost share of input i representing labor (L), capital (K), energy (E) and

material (M) and the ϵ_i represent the error in the ith equation. We assume that $E\epsilon_i=0$ and $E\epsilon_i\epsilon_j = \Sigma_{ij}$ for all i and j = L, K, E and M. The cost minimization hypothesis imposes the following set of exact restrictions on the parameters of the CID: (I) $\sum_j \beta_{ij} = 0$; (II) $\beta_{ij} = \beta_{ji}$ for all $i \neq j$ (III) $\sum_i \alpha_i = 1$ and $\sum_i \beta_{ij} = 0$.

The restrictions (I) to (III) are commonly known as homogeneity, symmetry and additivity constraints on the CID functions.

The additivity constraints are easily incorporated into equations (1) by dropping one of the share equations. We shall follow this convention by dropping the material input equation and wiring the remaining 3 equations compactly as:

$$(2) \qquad y = (I \otimes X)\,\beta + \epsilon$$

where y is a 3nx1 vector of n observations on the input cost shares with $y' = (S_{L1}, S_{L2}, \ldots, S_{Ln}, S_{K1}, S_{K2}, \ldots, S_{En})$, X is an nx5 matrix of observations on variables on the right-hand side of equation (1) with the t-th row $X_t = (1, \ln w_{Lt}, \ln w_{Kt}, \ln w_{Et}, \ln w_{Mt})$, β is a 15x1 vector of parameters of the CID equations with $\beta' = (\alpha_L, \beta_{LL}, \beta_{LK}, \beta_{LE}, \beta_{LM}, \alpha_K, \beta_{KL}, \ldots, \beta_{EM})$ and ϵ is a 3nx1 vector of observations on the errors. The vector ϵ is assumed to be distributed with 0 mean and covariance matrix $E\epsilon\epsilon' = \Sigma \otimes I$ where $\Sigma = ((\Sigma_{ij}))$ for i,j = L, K, E and M. The stochastic constraints of the homogeneity and symmetry constraints (I) and (II) can be compactly written as:

$$(3) \qquad R\beta = u$$

where R is a 6x15 matrix whose elements are specified by the homogeneity and symmetry conditions, β is a 15x1 vector of coefficients defined above and u is a 6x1 disturbance vector such that Eu = 0 with $Euu' = \Phi$ where Φ is a positive definite matrix. We follow Ilmakunnas (1986) in using a convenient parameterization of Φ such that $\Phi = \sigma_R^2 I$; the parameter σ_R^2 represents the degree of stickiness towards homogeneity and symmetry. As σ_R^2 approaches zero the stochastic constraints tend to become exact constraints.

The MR estimator of β in (2) under stochastic constraints (3) is given by

$$(4) \qquad b = \left(\tilde{\Sigma}^{-1} \otimes X'X + \frac{1}{\sigma_R^2} R'R\right)^{-1} (\tilde{\Sigma}^{-1} \otimes X')\, y$$

where $\tilde{\Sigma} = (Y - XB)'(Y - XB)/n$ is a consistent estimation of Σ, Y is an nx3 matrix with t-th row (S_{Lt}, S_{Kt}, S_{Et}) and B is a 5x3 matrix of coefficients of cost-shares equations for L, K, and E. We estimate b's using an iterative method until the estimated values converge (see Berndt and Wood, 1975).

The asymptotic variance-covariance of b is given by

$$(5) \qquad V(b) = (\tilde{\Sigma}^{-1} \otimes X'X + 1/\sigma_R^2\, R'R)^{-1}$$

It is easily verified that the covariance matrix (5) is also the mean square or risk matrix of b since the stochastic constraints are assumed to hold on the average (cf. Judge et al., 1985, pp. 58-59).

The estimates of σ_{ij} can be obtained from the formulas:

$$(6) \qquad \tilde{\sigma}_{ij} = (b_{ij} + \bar{S}_i\bar{S}_j) / \bar{S}_i\bar{S}_j \quad \text{for } i \neq j$$

and

$$(7) \qquad \tilde{\sigma}_{ii} = b_{ii} + \bar{S}_i^2 - \bar{S}_i/\bar{S}_i^2$$

where $\bar{S}_i$ and $\bar{S}_j$ are average values of cost shares for inputs i,j = L, K, E, M and b_{ij}'s are the MR estimates of β_{ij} in the CID equations (1). The asymptotic standard errors of σ_{ij}'s may be obtained from

$$(8) \qquad SE(\tilde{\sigma}_{ij}) = [V(b_{ij})/\bar{S}_i^2\,\bar{S}_j^2]^{1/2}$$

3. THE BOOTSTRAP IDEA AND MR ESTIMATES

3.1 The Bootstrap Idea

The bootstrap is a distribution-free method of determining the accuracy of the parameters of a model. The bootstrap theory is discussed in detail by Efron (1979, 1982). A survey of the bootstrap theory and applications is provided by Efron and Tibshirani (1986).

The bootstrap standard error can be used for calculating standard confidence intervals (CI) of σ_{ij} from formula $\tilde{\sigma}_{ij} \pm t_{df}^{\alpha}\, SE(\tilde{\sigma}_{ij})$ where $\tilde{\sigma}_{ij}$ is an estimator of the parameter σ_{ij}, $SE(\tilde{\sigma}_{ij})$ is the bootstrap SE of $\tilde{\sigma}_{ij}$, and t_{df}^{α} is the 100α percentile point from the t-distribution.

The bootstrap idea in the context of standard regression model $E(y_i) = \beta_o + \sum_{j=1}^{p} \beta\, X_{ij}$ may be described as follows. Suppose we have n observations on the dependent variable y and the regressors $(x_1, x_2, \ldots, x_p)$. Further, suppose that the regression errors $\epsilon_i = y_i - E(y_i)$ for i = 1, 2, ..., n are from an unknown distribution F and that b's are least squares estimators of β's. Then the bootstrap idea is to approximate an unknown distribution G(F) of $b_j - \beta_j$, by $G(F_e)$ where F_e is the empirical distribution of F for a given sample data set on the dependent variable and its regressors.

Now, consider a large number of random samples of size n with replacement, drawn from a box containing the least squares residuals $e_1, e_2, e_3, \ldots e_n$. Suppose one such sample is then

designated e_1^*, e_2^*, e_3^*, ..., e_n^* effectively yielding the "pseudo data" for y from $y^* = b_c + \sum_{j=1}^{p} b_j X_{ij} + e_i^*$ (i=1,2,...,n). This "pseudo data" along with the sample observations on the regressors would then constitute a set of sample values for the bootstrap empirical distribution.

In view of the fact that the least square residual e's are not independent even though the ϵ's have this property and that the e's are a bit smaller than the ϵ's, the bootstrap sampling from e's can be downward biased. This bias can be reduced by scaling up the e_i's by a factor of $[n/(n-p-1)]^{1/2}$ (see Freedman and Peters, 1984). We used a scaling factor in our bootstrap results reported below.

3.2 The Results

The MR estimates of σ_{ij} were obtained for three values of the stickiness parameter σ_R^2 : $\sigma_R^2 = 10^{-4}$, $\sigma_R^2 = 10^{-6}$ and $\sigma_R^2 = 10^{-8}$.

However, we shall present the detailed results for $\sigma_R^2 = 10^{-4}$ only in view of space limitations. Moreover, the hypotheses that energy and capital are complements was found not to be violated when $\sigma_R^2 = 10^{-6}$ and $\sigma_R^2 = 10^{-8}$. The MR estimates corresponding to $\sigma_R^2 = 10^{-8}$ correspond to the exact constraints case. The point and interval estimates of σ_{ij}'s for $\sigma_R^2 = 10^{-4}$ are given in Table 1.

The results in Table 1 show that the estimates of σ_{KE} and σ_{EK} are of opposite sign; thus the energy-capital can be substitutes instead of complements when the stickiness parameter is equal to $\sigma_R^2 = 10^{-4}$. But, the asymptotic standard error of $\hat{\sigma}_{KE}$ in column 3 and the bootstrap standard error in column 5, which are the parametric and non-parametric measure of the accuracy of the estimator σ_{ij}, respectively , are large. Therefore, it might be worthwhile to calculate the 75% CI to determine if the positive value of $\hat{\sigma}_{KE}$ is included in the CI. The possibility of a positive value in the CI would suggest that the hypothesis of energy and capital substitutability cannot be rejected at the 25% level. The 75% CI_a in column 4 and 75% CI_b in column 6 represent the parametric confidence intervals with $SE(\sigma_{ij})_a$ and $SE(\sigma_{ij})_b$, respectively. These 75% CIs appear to include a posi-

Table 1: The Mixed Regression Estimates of Allen Partial Elasticity of Substitution

σ_{ij} when $\sigma_R^2 = 10^{-4}$

	$(\hat{\sigma}_i)$	$SE(\hat{\sigma}_{ij})_a$	75% CI_a	$SE(\hat{\sigma}_{ij})_b$	75% CI_b
1) σ_{LL}	-1.607	0.128	[1.694, -1.520]	0.103	[-1.677, -1.537]
2) σ_{LK}	1.174	0.547	[0.803, 1.546]	0.476	[0.851, 1.497]
3) σ_{LE}	1.427	0.828	[0.865, 1.989]	0.464	[1.112, 1.742]
4) σ_{LM}	0.488	0.132	[0.398, 0.578]	0.088	[0.428, 0.548]
5) σ_{KL}	0.770	0.424	[0.482, 1.058]	0.336	[0.542, 0.998]
6) σ_{KK}	-6.391	2.095	[-7.813, -4.968]	2.141	[-7.844, -4.938]
7) σ_{KE}	0.941	3.660	[-1.544, 3.426]	2.761	[-0.934, 2.816]
8) σ_{KM}	0.322	0.524	[-0.034, 0.677]	0.439	[0.024, 0.620]
9) σ_{EL}	1.009	0.363	[0.762, 1.255]	0.308	[0.800, 1.218]
10) σ_{EK}	-3.444	1.556	[-4.500, -2.388]	1.617	[-4.542, -2.346]
11) σ_{EE}	-12.260	3.356	[-14.539, -9.981]	3.194	[-14.429,-10.091]
12) σ_{EM}	0.491	0.431	[0.198, 0.784]	0.394	[0.224, 0.758]

Notes: $\hat{\sigma}_{ij}$: The point estimate of the Partial Allen elasticity of substitution of factor input i with j.

$SE(\hat{\sigma}_{ij})_a$: The standard error of $\hat{\sigma}_{ij}$ from the asymptotic formula.

75% CI_a: The 75% confidence intervals with $SE(\hat{\sigma}_{ij})_a$ and $t_{60}^{75} = .679$.

$SE(\hat{\sigma}_{ij})_b$: The standard deviation of the bootstrap distribution.

75% CI_b: The 75% confidence intervals with $SE(\hat{\sigma}_{ij})_b$ and $t_{60}^{75} = .679$.

tive value of $\hat{\sigma}_{KE}$. Hence, the hypothesis that energy and capital may be substitute appear not to be rejected by the data.

How does the existence of fat-tailed errors affect the CI? We investigated this question by reestimating the σ_{ij}'s by the MR method with fatter-tailed errors. These errors were obtained as described below. Under the assumption that residuals from (2) are normally distributed, we generated a fatter-tailed error by mixing two sets of normally distributed errors such that a proportion (1-s) of original errors with 0 mean and covariance matrix $\Omega = \Sigma \otimes I$ were combined with a proportion s of another set of errors with 0 mean and covariance matrix $d\Omega$ where d is a scalar greater than 1. The matrix of errors so obtained are distributed with 0 mean and covariance matrix $\Omega(d) = (1-s)\,\Omega + s(d\Omega)$ and these errors are fatter-tailed than the original errors (see Flood et al., 1984). We used d=4 and s=2/25 such that the Kurtosis coefficient of fat-tailed errors is about 1.64. The use of fatter-tailed errors resulted in somewhat higher $SE(\hat{\sigma}_{ij})_b$ compared to those in Table 1. The 75% CI were also computed and again failed to reject the substitutability of energy and capital for $\sigma_R^2 = 10^{-4}$.

4. SUMMARY REMARKS

In this paper we examined the sensitivity of the estimates of the partial Allen elasticity of substitutions and their confidence intervals when homogeneity and symmetry hypotheses hold stochastically compared to exactly.

A simple parameterization of the covariance matrix for the disturbance term in the stochastic constraints was considered. It was shown that for some a priori specification of σ_R^2 (i.e., $\sigma_R^2 = 10^{-4}$) where σ_R^2 has an interpretation as a coefficient of stickiness towards homogeneity and symmetry yield a positive estimate of σ_{KE}. The 75% CI of σ_{KE} were computed using both asymptotic SE and bootstrap SE and they failed to exclude a positive value for σ_{KE}. Therefore the hypothesis that energy-capital may be substitutes cannot be rejected at the 25% level. These results might be interpreted in terms of the rationale given by Berndt and Wood (1979) in terms of capital utilization.

FOOTNOTES

[1] One reason these avenues have failed to resolve the controversy might be that their studies use aggregate data which makes it difficult properly to capture the general equilibrium effects of an energy price shock on business (see Solow, 1987).

[2] Estimates of the parameters of the cost function under stochastic constraints can be carried out in either a mixed regression or Bayesian framework. In this paper we will focus on the MR approach.

[3] The Mixed Regression model is a convenient econometric technique for combining information from a given sample with prior non-sample stochastic information with a view to obtaining a more efficient estimate of regression coefficients. The MR model has proven to be useful if judged solely by the plausibility of the results obtained from it, although the assumptions it is based on are somewhat logically flawed (see Zellner, 1975). This method was originally proposed by J. Durbin in 1953 and later developed more fully by Theil and Goldberger (1961) on heuristic grounds. However, it can also be interpreted as a Bayes estimator and has been applied in areas of consumer demand (e.g., see Paulus, 1975) and cost functions (e.g., see Illmakunnas, 1986 and references therein).

[4] Efron (1982) has provided some evidence for the relative performance of the jackknife and bootstrap methods. He found that while both jackknife and bootstrap standard errors provide an almost unbiased estimate of the parameters, the bootstrap method has a lower coefficient of variation than the jackknife method.

REFERENCES

Berndt, E.R. and D.O. Wood (1975), "Technology, prices and the derived demand for energy," Review of Economics and Statistics, 57:259-268.

Berndt, E.R. and D.O. Wood (1979), "Engineering and econometric interpretations of energy-capital complementarity," American Economic Review, 69:342-354.

Christensen, L.R., D.W. Jorgenson and L.J. Lau (1971), "Conjugate Duality and the Transcendental Logarithmic Production Function," Econometrica, 39:255-256.

Efron, B. (1979), "Bootstrap Methods: Another Look at the Jackknife," Annals of Statistics, 7:1-26.

Efron, B. (1982), "The Jackknife, the Bootstrap and Other Resampling Plans (Philadelphia: Society for Industrial and Applied Mathematics).

Efron, B. and R. Tibshirani (1986), "Bootstrap Methods for Standard Errors, Confidence Intervals and Other Measures of Statistical Accuracy," Statistical Science, 1:54-77.

Solow, J.L. (1987), "The Capital-Energy Complementarity Debate Revisited," American Economic Review, 77:605-614.

Taylor, W.E. (1983), "Finite Sample Distribution Theory," Econometric Reviews, 2:1-84.

Theil, H. and A.S. Goldberger (1961), "On Pure and Mixed Statistical Estimation in Economics," International Economic Review, 2:65-78.

Theil, H. (1971), Principles of Econometrics (New York, Wiley).

Tsurumi, H., H. Wago, and P. Ilmakunnas (1986), "Gradual switching multivariate regression models with stochastic cross-equational constraints and an application to the KLEM translog production model," Journal of Econometrics, 31:235-253.

Zellner, A. (1975), "Bayesian approach and alternatives in econometrics," in Fienberg, S.E. and A. Zellner (eds.), Studies in Bayesian Econometrics and Statistics (North Holland, Amsterdam), pp.39-54.

Flood, L.R., R. Finke and H. Theil (1984), "Maximum Likelihood and Minimum Information Estimation of Allocation Models with Fat-Tailed Error Distribution," Economics Letters, 16:213-218.

Freedman, D.A. and S.C. Peters (1984), "Bootstrapping an Econometric Model: Some Econometric Results," Journal of Business and Economic Statistics, 2:150-158.

Griffin, J.M. and P.R. Gregory (1976), "An Intercountry Translog Model of Energy Substitution Responses," American Economic Review, 66:845-857.

Ilmakunnas, P. (1986), "Stochastic Constraints on Cost Function Parameters: Mixed and Hierarchical Approaches," Empirical Economics, 11:69-80.

Judge, G.G., W.E. Griffiths, R.C. Hill, and T-C Lee (1985), The Theory and Practice of Econometrics. (New York: Wiley).

Pindyck, R.S. (1979), "Interfuel Substitution and the Industrial Demand for Energy: An International Comparison," Review of Economics and Statistics, 61:169-179.

IV. STATISTICAL GRAPHICS

Dimensionality Constraints on Projection and Section Views of High Dimensional Loci
George W. Furnas, Bell Communications Research

A Demonstration of the Data Viewer
Catherine Hurley, University of Waterloo

Visualizing Multi-Dimensional Geometry with Parallel Coordinates
Alfred Inselberg, IBM Scientific Center and University of Southern California; Bernard Dimsdale, IBM Scientific Center

On Some Graphical Representations of Multivariate Data
Masood Bolorforoush, Edward J. Wegman, George Mason University

Graphical Representations of Main Effects and Interaction Effects in a Polynomial Regression on Several Predictors
William DuMouchel, BBN Software Products Corporation

Dimensionality Constraints on Projection and Section Views of High Dimensional Loci

George W. Furnas
Bell Communications Research

Abstract

Fundamental limitations are presented for two general graphical techniques for constructing geometric views of high-dimensional loci, *projection* and *section*. *Projections* can only easily display aspects of structure that are of low *dimensionality*. *Sections*, i.e, intersections of affine subspaces with a locus, can easily display structure of only low *co-dimensionality* (and hence high dimensionality). However, compositions of section and projection can display aspects of structure of any intermediate dimensionality. These assertions are proven for fundamental idealization of loci that are arbitrary affine subspaces of a high-dimensional space. The issues introduced by finite extent, by curvature, by quantization and by error noise are then discussed, basically in terms of notions of scale. Examples of using the composition technique are given, examining the structure of two high-dimensional objects embedded in a six-dimensional space.

1. Introduction

The investigation of high dimensional loci arises in both mathematics and statistics. In mathematics, sets of equations and inequalities, or computational procedures can define mathematical objects of high dimensionality. Graphics provides one set of tools to augment algebraic attempts to understand the structure of these objects. The typical graphical approach is to make various 2-dimensional projections and sections of the locus, from which some sense of its structure is obtained.[1] [2] [3] In statistics, multivariate data form high dimensional point-clouds whose structure must be detected and modeled. Again graphics are playing an increasing role in augmenting parametric characterization of the structure of such loci, particularly in the exploratory stages of data analysis[4] [5] [6] [7] [8] [9] [10] [11] [12]. Though statisticians sometimes use various glyph variation schemes[1] for graphical presentation of multivariate data (e.g., Chernoff Faces[13], trees and "castles"[14]), geometric transformations, usually projection, are also used to produce two-dimensional renditions of high dimensional loci (e.g., [4] [8] [11]). This paper represents a basic attempt to understand the theoretical power of views generated by such geometric transformations.

1.1 A Motivating Example: The 4-point Ultrametric Locus

The inherent limitations of low-dimensional projections can be illustratied by the 4-point Ultrametric Locus, a particular mathematically defined locus embedded in 6-space. (This locus arises in efforts to understand the families of distance matrices satisfying different metrics, e.g., general metric, euclidean, ultrametric.[15] Its importance here is simply that it is an interesting locus in six-space.)

Consider, for four points A, B, C and D, the vector of six pair-wise distances between them:

$$(u, v, w, x, y, z) = (d_{AB}, d_{AC}, d_{BC}, d_{AD}, d_{BD}, d_{CD}).$$

Of all possible (non-negative) sextuples of such distances, consider only those that correspond to distances satisfying the Ultrametric Inequality:

$$d_{ij} \leq max(d_{ik}, d_{jk}) \qquad i,j,k \in \{A, B, C, D\}$$

Ultrametric distances are interesting because there is a one-to-one correspondence between such sextuples of distances and hierarchical clusterings of objects A, B, C and D, or equivalently rooted ultrametric trees with these four objects as their leaves. Thus understanding this locus amounts to understanding the complete set of Ultrametric trees on 4-points.

The set of ultrametric sextuples forms a locus (UM-Locus) of some type embedded in the six dimensional space of all sextuples. For various algebraic reasons, this locus was known to have interesting structure. To get a better sense of it in detail, one might try to "look" at it using a powerful high-dimensional rotation and projection system, such as *The Data Viewer*, developed by Andreas Buja and his colleagues [12] for looking at high dimensional multivariate point-clouds. To do this, a point-cloud representation of the locus was created by generating and testing each point in the six-dimensional unit hypercube whose coordinates were multiples of 0.10. Points on this grid that satisfied the Ultrametric Inequality were collected, and the rest ignored. The resulting six-dimensional point-cloud was then entered into *The Data Viewer* which then dynamically rotated the locus and generated a continuous moving sequence of two-dimensional projections. One such projection is shown in Figure 1.

Author's Address: Bell Communications Research, Inc., 445 South Street, Room 2M-397, P.O. Box 1910, Morristown, New Jersey, 07960-1910 USA

The author would like to thank Andreas Buja, John Schotland, and Adolfo Quiroz for their comments on drafts of this paper and its related proofs.

1. Such schemesuse some "glyph", such as an iconic face, whose various graphical features (e.g., aspect ratio of the face, size of eyes, etc.) are parameterized and associated with variables. Thus a set of points becomes a family of glyphs.

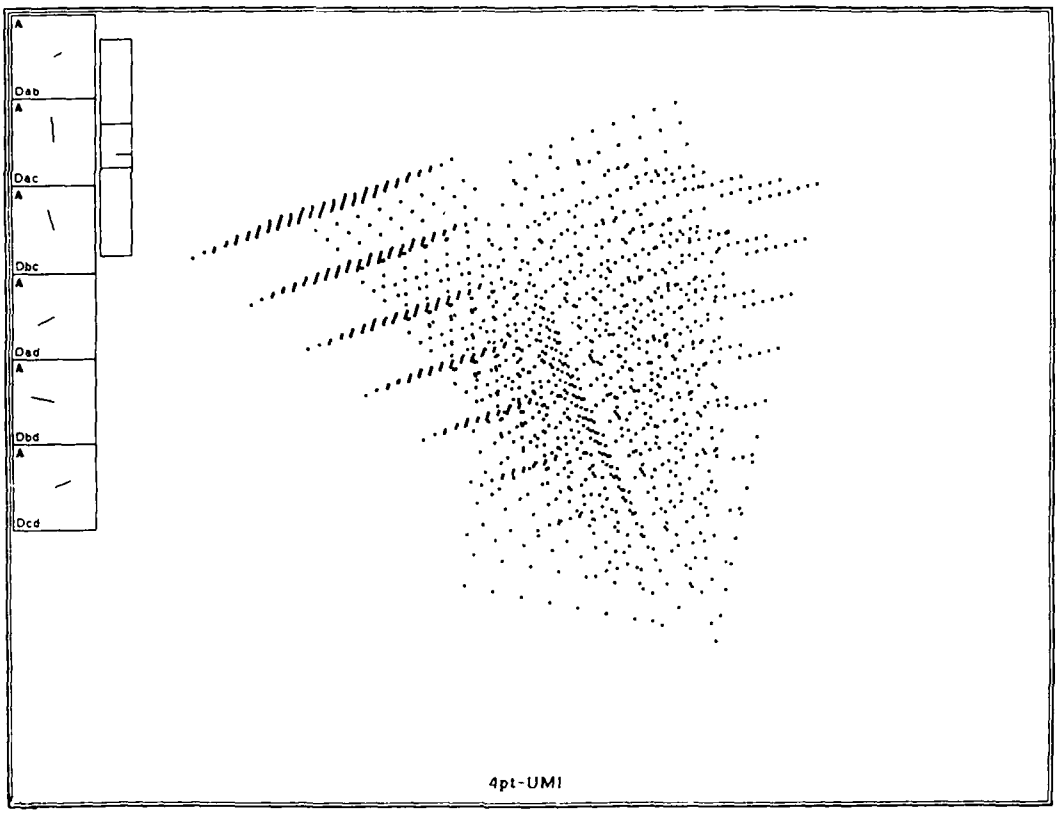


Figure 1. 2-Dimensional projection of the 6-dimensional, '4-Point Ultrametric Locus'

The critical feature of Figure 1 is that it shows essentially nothing interesting! The only visible aspects of the structure are artifactual: e.g., the edges and corners seen in the figure are the edges and corners of the hypercube that the locus was sampled from and not special features intrinsic to the structure itself.

Here is a graphical tool used frequently by statisticians, sometimes with marked success, to look at high dimensional loci. Yet for this locus, which is known to have interesting structure, projection shows nothing. The work presented in this paper represents an attempt to understand what is happening in Figure 1, by addressing a simple, though fundamental idealized case.

1.2 The Two Geometric Viewing Techniques

We will actually investigate two common geometric techniques for deriving a low dimensional picture of a higher dimensional locus: *projection* and *section*.

By a *k–projection* we will mean an orthogonal projection of a high dimensional locus embedded in n-space onto a k-dimensional affine subspace (e.g., onto a line, plane, or general k-dimensional hyperplane, not necessarily through the origin. Orthogonality is with regard to the canonical inner product on $\mathbb{R}^n$.) Most typically, this means projecting from n-space onto some 2-dimensional plane at some orientation in the n-space. This 2-projection is used as a 2D graphic, i.e., a picture on paper or in a video display.

By a *k–section* we will mean the intersection of a k-dimensional affine subspace with the high dimensional locus residing in n-space. A 2-section arising from intersecting some plane in the n-space with the locus can be presented as a sort of cross-sectional picture of the locus.

Although for simple graphics k=2, interest in the general case of k>2 is not just theoretical. There are ways to present graphics that are more than 2-dimensional, e.g., using stereo presentation, color and motion/time (e.g., [4] [2] [11] [12] [5] [6] [7] [16]). The results that follow should pertain to these higher-dimensional graphics as well. Also, as will be seen, it will be useful to consider the composition of section and projection operations of various dimensionalities, and their net effect can only be understood by considering the general case.

2. The Affine Subspace Idealization

Imagine that a demon opponent presents an investigator a n-dimensional black box that has a target object embedded in it, and challenges the investigator to use geometric/graphical techniques to discover what is inside. The demon's goal is to put in something hard; the investigator's -- to figure what is there anyway. This fully general problem is exceedingly difficult, so we consider here a fundamental simple case: Suppose we allow the demon to

put only certain very simple high-dimensional loci in the box: *flats*. By a *flat*, we will mean an arbitrary affine subspace: a point, line, plane, or hyperplanes (not necessarily through the origin). In particular, an *m-flat* will mean an m-dimensional affine subspace embedded in n-space.

Note that these special loci differ from loci of practical interest in several ways. They are infinite in extent and high translational symmetry (a line looks the same everywhere along its length). In addition, unlike statistical loci (and some mathematical ones in theory, and many in computational practice), they are continuous. In this difference resides the idealization: and we will try to return across this gap at the end. In any case, flats are sufficiently primitive and fundamental objects that understanding their behavior has value in its own right.

Accepting for the while this restriction, the situations is thus: the demon will put some target m-flat in the n-space, and the investigator will try to use k-projection or k-section to look at what is there. What will the investigator see?

2.1 Constraints on Projection Views

Consider first the case of *projection*, i.e., "How does an m-flat appear in a k-projection?" The answer turns out to be quite simple:[2]

> *The operation of* **k-projection** *will yield an image of the* **m-flat** *that* **almost surely**
> - *preserves dimensionality of a* **m-flat** *(i.e.,* **m-flat** *in n-space ==>* **m-flat** *in k-space), when* $\mathbf{m < k}$, *and*
> - *is of full* **k** *dimensionality when* $\mathbf{m \geq k}$ *(thus indistinguishably covering the k-dimensional viewing space).*

Thus for example, a point (0-flat) in 3-space always appears as a point (0-flat) in a 2-projection. Thinking of the casting of a shadow as a projection, recall that the shadow of a point is a point, regardless of its position in 3-space. It is likewise true in n-space. Similarly, a line (1-flat) in 3-space will *almost surely* appear as a line in a 2-projection; the shadow of a line is *almost surely* a line. The italicized phrase, *almost surely*, is being used in the technical (measure theoretic) sense.[3] That is, for example, it is possible for a line (1-flat) to 2-project not into a line (1-flat) but into a point (0-flat). However, this can happen only in the singular case that line is perpendicular to the 2-flat used for the projection (the viewing space). This singular case has measure zero (i.e., zero probability if flats are chosen randomly), and hence *almost surely* the 2-projection of a 1-flat is a 1-flat.

2. Proofs of the *almost surely* assertions about dimensionality of projections and sections will be published elsewhere, and are also available in [17].

3. The *almost surely* statements here require only that underlying probability distributions be absolutely continuous w.r.t. the Lebesque measure on the corresponding natural euclidean parameter spaces. For example coordinates of the $n \times (n-p)$ matrix defining a p-dimensional linear subspace could be sampled from the standard spherical multivariate normal on $\mathbb{R}^{n(n-p)}$. See the proofs for details.

The projection operation cannot preserve dimensionality of a target m-flat if m gets so large that it exceeds the dimensionality of the viewing space. Illustrating this case where $m \geq k$, note that a plane (2-flat) in 3-space will *almost surely* 2-project onto the whole projection plane. The whole 3-space (3-flat) will also 2-project to cover the whole plane. The 2-projection alone cannot distinguish a 2-flat target from a 3-flat one.

Thus if the demon puts a point or a line in the box, the investigator can easily disclose it with an arbitrary 2-projection, and thereby win. However if the demon sets as a target a higher dimensional m-flat, all 2-projections will be completely and indistinguishably covered.

A second look at the Ultrametric Locus of Figure 1 bears out the results just given. The projection made visible only 0-dimensional features (point-like corners) and 1-dimensional features (line-like edges) of the locus. Unfortunately these were artifactual aspects of the locus. The interesting structure apparently was in the higher dimensionality, and to the demon's gratification, was self-obscured by the projection operation.

This means that projection is a powerful technique for identifying low dimensional affine substructures in high dimensional space, but *almost surely* useless in finding higher dimensional ones. Put another way, if the affine structure of interest is of low dimensionality, essentially ANY projection will show it clearly. If it is of high dimensionality (where "high" is often only $m > 2$, since typical projections are 2D), only very singular projections will show it. It is the struggle against this *almost surely* condition that makes the pursuit of informative projections (e.g., in Projection Pursuit [18]) so difficult.

2.2 Constraints on Section Views

Fortunately for the investigator, the second tool available for creating low-dimensional views, section, has a complementary power. Considering the case of *section*, we ask, "How does an m-flat appear in a k-section?"

The answer to this requires the notion of *co-dimension*. The co-dimension of a flat is the complement of its dimensionality with respect to that of the full space. That is, in n-space, the co-dimensionality of a m-flat is defined to be $i \equiv n-m$. Thus the co-dimensionality of a plane in 3-space is (3-2)=1, that of a point in a plane is (2-0)=2.

Whereas the effect of projection was put simply in terms of dimension, the effect of section is put simply in terms of co-dimension:

> *The operation of* **k-section** *will yield an image of an* **m-flat** *that* **almost surely**
> - *preserves the co-dimensionality of a* **m-flat** *(i.e.,* **(n-i)-flat** *in n-space ==>* **(k-i)-flat** *in k-space), when* $\mathbf{(n-m) < k}$, *and*
> - *is empty (i.e., indiscriminately missing m-flats) when* $\mathbf{(n-m) \geq k}$.

Let i be the co-dimension of the m-flat, i.e., $m=(n-i)$. If $i \equiv (n-m) < k$, then the $(n-i)$-flat *almost surely* appears as a $(k-i)$-flat in the k-section. Thus for example, in 3-space a

line [=(3-2)-flat] *almost surely* appears as a point [=(2-2)-flat] in an arbitrary 2-section. A plane [=(3-1)-flat] *almost surely* appears as a line [=(2-1)-flat] in an arbitrary 2-section.

On the other hand, if $i \equiv (n-m) > k$, it *almost surely* disappears from the k-section. Thus in three space an arbitrary 2-section will *almost surely* miss a target point [=(3-3)-flat]. It will reveal the point only under the singular condition that the viewing plane happens to be positioned and oriented so as to pass through the point.

Thus if the demon puts a flat of co-dimension 0, 1, or 2, (dimensionality n, $n-1$ or $n-2$) in the box, investigator can easily disclose it with an arbitrary 2-section. But if the demon sets a target of higher co-dimension, all the investigators' 2-sections will *almost surely* miss. So, whereas projection is a powerful technique for identifying structure of low dimension, section is useful for finding structure of low co-dimension.

2.3 Complementarity and Composition

These previous properties of projection and section are summarized in Figure 2 for flats in 6-space, the dimensionality of the black box containing the Ultrametric Locus of Figure 1. If the affine structure of interest is of low co-dimensionality, essentially ANY projection will show it clearly. If it is of low co-dimensionality essentially ANY section will show it. Thus these two patterns of strengths are complementary. However, even the union of the two techniques is still limited. Given that one seeks only two-dimensional pictures, so that $k=2$, projection can find substructures of dimensionality 0 and 1, and section can find dimensionality n, $n-1$, and $n-2$. In cases where $n \leq 4$, this covers all the cases. But for larger n, there is a gap between the low dimensional and low co-dimensional extremes, and so the demon can still win.

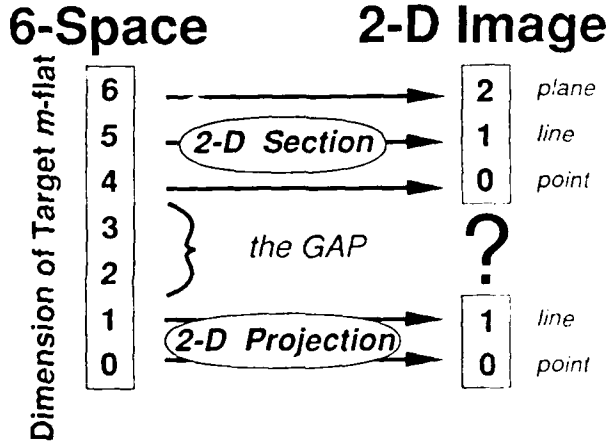


Figure 2. The joint capabilities of section and projection.

Fortunately, composition of these techniques can completely bridge the gap. For example, consider the problem of finding a 3-flat in 6-space, using a $k=2$ dimensional viewing space. Neither single approach will find it: Since $m=3 > k=2$, it will *almost surely* indiscriminately cover 2-projections, and since $n-m=6-3=3 > k=2$, it will *almost surely* not appear in 2-sections. How can an informative 2D view be created?

Following Figure 3, note that a 4-section of the 6-space will *almost surely* contain an image of the 3-flat, since $n-m=6-3=3 < k=4$. Since section preserves co-dimension, the (6-3)-flat will become a (4-3)-flat [=1-flat] in the 4-section. Thus we have a section that at least contains some image of the target. The problem is that a 4-section is not a 2D picture. That is easily solved by taking a 2-projection of the 4-section. The 4-section is now a new 4-dimensional black box with a 1-flat target in it. Correspondingly let $n' \equiv 4$, $m' \equiv 1$ and $k' \equiv 1$ Thus, since $m'=1 < k'=2$, dimensionality will be preserved by the projection, yielding a clearly visible 1-flat (line) in the final image. That is, if the investigator takes a 2-dimensional projection of a 4-dimensional section of a 6-dimensional space, and sees a line, she has just found a 3-flat.

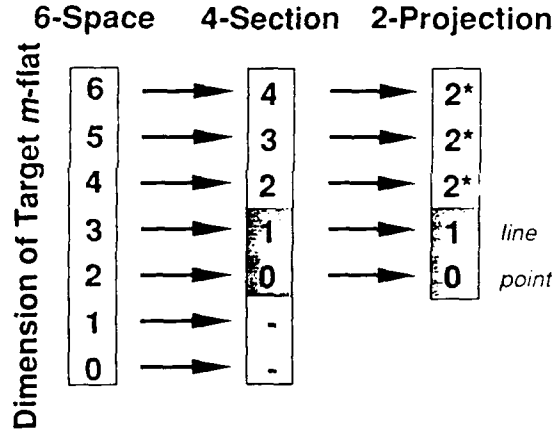


Figure 3. Effects of a 4-section followed by a 2-projection, and the remaining gap.

By similar combinations, the investigator can reveal an arbitrary m-flat. E.g., it will *almost surely* appear as a line in a 2-projection of a $(n-m+1)$-section. Equivalently, the m-flat target could be revealed by an alternative composition, taking a 2-section of a $(m-1)$-projection. In either case the investigator can now always beat the demon.

It should be stressed that when used as suggested by the constraints discussed here, m-flat structure can be found WITHOUT SEARCH through the orientation and location parameters of the section and projection operations. The "*almost surely*" considerations mean that sections and projections of *arbitrary* positions and orientations should yield the desired result. One must only examine at most n/k k-dimensional views. Each corresponds different dimensionalities of the initial j-section ($j = n, n-k, n-2k, n-3k, \ldots, k$), which precedes the final k-projection in the composite strategy.

3. Two Examples

The previous theory is based on the idealization that the demon can only use flats as targets, yet real loci can deviate from this idealization in many ways. Before considering such deviations in detail, we will first present some examples to show that the technique holds promise even for real loci.

Views of the loci in this section were again generated using *The Data Viewer* to do 2-D projections and systematically using its "brushing" facility compositely to do sections. (Note that a p-dimensional brush, i.e., conditioning on a linear combination of p variables leaves $n-p$ free to vary, creating a $(n-p)$-dimensional section).

3.1 Example 1: The 4-point Ultrametric Locus revisited

In this example the composition of section and projection is used to get a more informative display of the Ultrametric Locus. An arbitrarily oriented 2-projection of the full locus was presented in Figure 1.

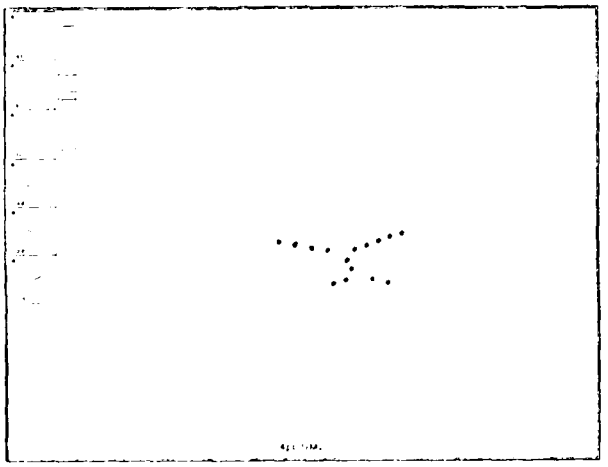


Figure 4. A 2-projection of a 4-section of the Ultrametric Locus.

Figure 4 presents a 2-projection of a 4-section through the locus (the same 2-projection as in Figure 1; so Figure 4 is actually embedded in Figure 1.) The result is a 5-segment tree structure. Essentially all such sections have this structure (Figure 5 shows another completely different section and projection.) These trees are made up of 1-flat pieces.

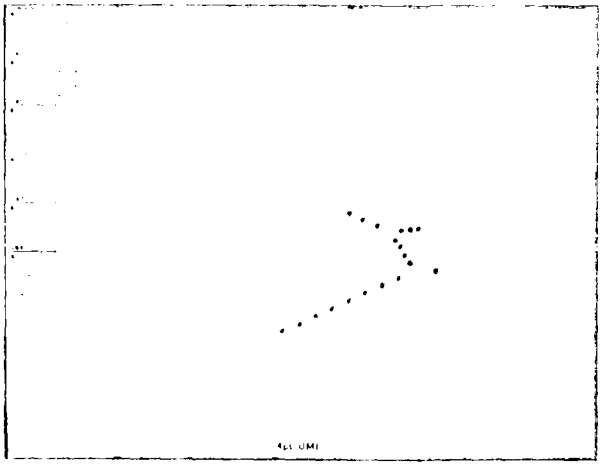


Figure 5. Another 2-projection of a 4-section of the Ultrametric Locus.

Working backwards through Figure 3, we can see that a 1-flat image in a 2-projection of a 4-section corresponds to a 3-flat in the embedding 6-space. That is, the locus is an articulated tree-like collection of 3-flat pieces. Further investigations can show that there are three distinct though connected sets of these 5-segment images.

All of these results are consistent with what is known about the set of ultrametric distances on four points. It has been mentioned that such distances correspond to distances in rooted binary trees on four points. There are 15 different such binary tree topologies, one associated with each of the segments of the three 5-segment shapes in the figures. Each of the 15 tree topologies has three continuous parameters that affect distance: the distance matrix is altered in a continuous fashion by changing the heights of the three internal nodes of the rooted binary tree. This explains the local three-dimensionality of the locus as revealed in the line-like appearence in the 2-projection of the 4-section of Figures 4 and 6. The composition of section and projection yields a powerful look at this articulated high dimensional object, even though it is not simply a flat.

3.2 Example 2: A three dimensional torus in 6 dimensional space

The second example examines a curved object, a three dimensional torus in 6 dimensional space. Such a torus is simply the Cartesian product of three circles. I.e., the set of sextuples (u, v, w, x, y, z) such that

$$\begin{aligned} u^2 + v^2 &= 1 \\ w^2 + x^2 &= 1 \\ y^2 + z^2 &= 1. \end{aligned}$$

Note that these three equations define a 3-manifold embedded in 6 space. This continuous object was turned into a point-cloud by taking 10 points around each circle. The Cartesian product thus yielded 1000 points. Figure 6 shows four simple 2-projections of the resulting toroidal cloud in 6 dimensional space. Note that beyond a general curved convex appearance, the special character of the structure is obscured in these simple projections.

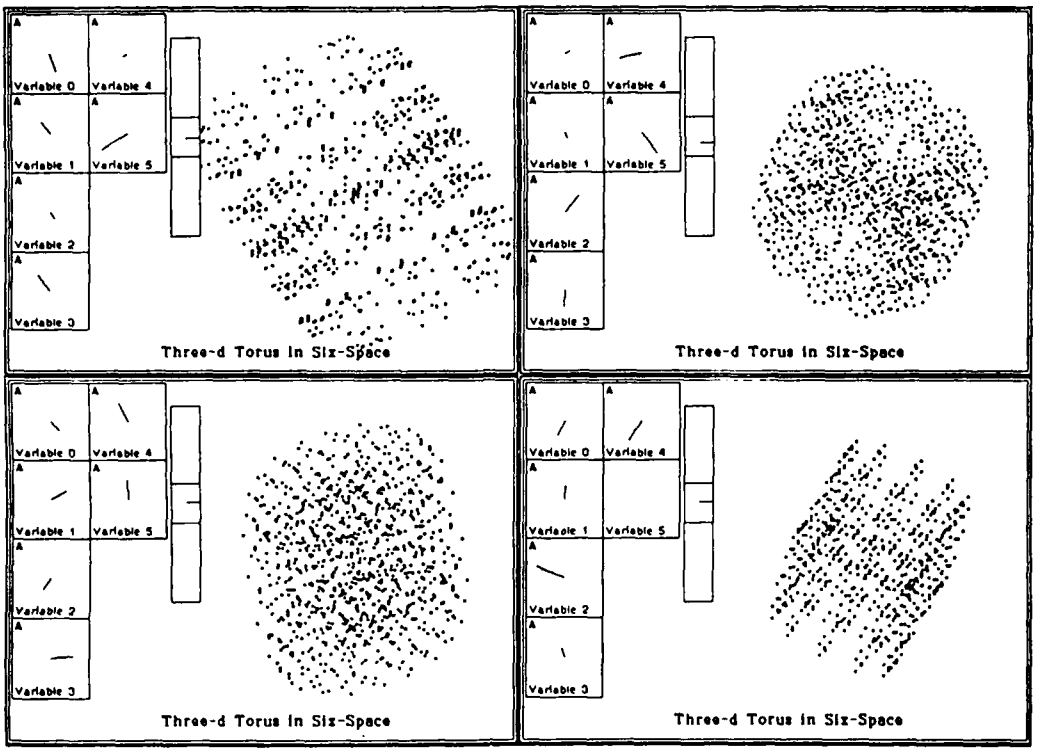


Figure 6. Four 2-projections of a 3-torus in 6-space.

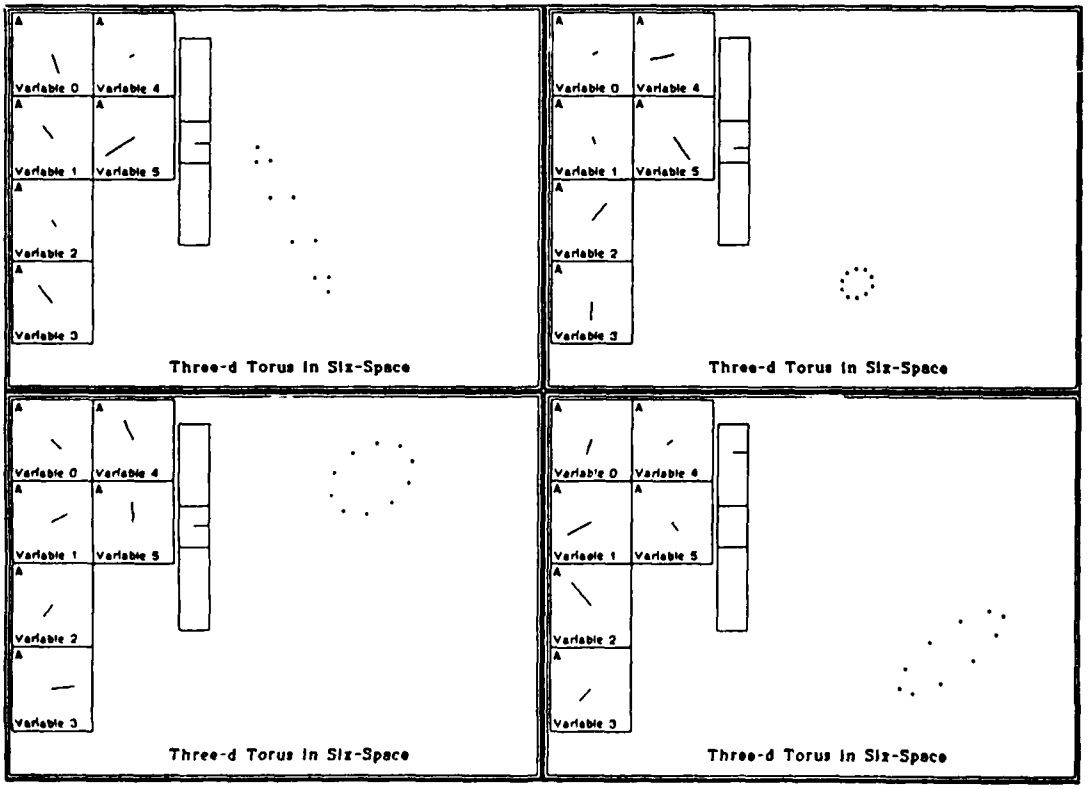


Figure 7. Four 2-projections of 4-sections of the 3-torus in 6-space.

Figure 7 shows four corresponding 2-projections-of-4-sections of the torus in 6 dimensional space. The fundamental circular structure is clearly visible and, referring back to the diagram of Figure 3, the appearence of the locus as curves in the viewing plane evidences its local 3-dimensionality.

4. Deviations from the Idealization

The previous two examples illustrate how the earlier ideal theory seems to extend to less ideal cases: both the limitations of sections and projections and the possible power of their composition are manifest. In both of these examples, as in many real situations, the high dimensional loci of interest differ in many ways from the idealized case of affine subspaces. In this section several of these deviations are discussed, with the principal conclusion that once a suitable level of scale is determined, ideal results remain useful, thanks to the robustness of linear approximations. The treatment here is casual and conjectural; other's efforts at formalization would be welcome.

4.1 Finite Extent

Many mathematical loci and (presumably) all empirical statistical ones are bounded *in extent*. To begin to understand the implications of boundedness, consider first the simple case of bounded pieces of m-flats. Since real viewing windows (paper or CRT screens) are also bounded, the relative scale of the target and viewing bounds is important. In particular, on sufficiently large scale (i.e., the target object is much smaller than the window), an m-flat-piece becomes point-like and projection will show it. On a sufficiently small scale in its neighborhood, the m-flat-piece becomes like an m-flat. Then the previous techniques of section and projection should work as described. Thus the viewing process requires an additional tool which can rescale the object with respect to the window size. Projection is first used at large scale to locate the object as a point-like entity. Then scale is reduced while staying centered on the object until the object looms large with respect to the window bounds, whereupon section and projection can be used.

4.2 Non-Linear Loci

4.2.1 Manifolds

Like the second example above, many interesting loci are manifolds that are curved, not flat. Technically, however, any manifold appears increasingly flat when viewed more and more locally. Thus the earlier results should hold with respect to the image of all local regions under section and projection. For example, the image of a 2-manifold in 3-space under 2-section should *almost surely* be locally 1-dimensional. But something which is locally 1-dimensional is a 1-manifold. Thus the results should generalize to manifolds, with a further caveat: The *almost surely* condition here means that sometimes there could occasional local alteration in dimensionality -- singularities can be introduced.

For a simple example, consider how best to tell a hollow sphere from a solid one. One is a 2-manifold in 3-space, the other a piece of a 3-flat in 3-space. Since the distinction between these is not one of low dimensionality; they will look the same in projection. They differ with respect to low co-dimensionality, so only section can distinguish them. One will appear as a ring (1-manifold) the other as a disc (piece of a 2-flat). One could by similar means distinguish hollow and solid hyper-spheres. Topographic maps are another interesting examples of the implicit application of this theory. The surface of a piece of terrain is essentially a 2-dimensional manifold in 3-space. Its structure is of co-dimension 1 and cannot be conveyed in a 2-D map by projection. Instead topographic contours, i.e., a family of 2-sections, display its shape -- as curves (structures of co-dimension 1) in the image plane.

4.2.2 Hyper-surfaces with singularities

The investigation of general m-surfaces, i.e., surfaces that may have singularities is more problematic. First note, though, that singularities are structures of lower dimensionality, p, where $p<m$. If one can partition the structure *a priori* into singularity-substructures by dimensionality, then each dimensionality can be examined according to the preceding treatment of manifolds. If no such partition is available *a priori*, the situation is more difficult. The problem is that structures of different dimensionality are present at the same time and can obscure each other. Singularities of dimension $p=m-1$ can be seen along with the m-surface by section and projection. They will appear as 0-dimensional singularities on a 1-manifold, e.g., like a cusp-point on a bifurcating curve. However if $p<m-1$, the singularities will either be lost by k-sections if k is small enough to clearly reveal the m-structure, and obscured by the over-projecting m-structure, if k is large enough not to miss the p-structure. A simple example of losing the singularities is the inability to see the exact location of mountain peaks (singularities of co-dimension $p=3=m-2$) in a topographic map. The 2-sections generating the contours *almost surely* miss the exact peak location -- hence the need for a special map symbol to mark them.

4.2.3 Intersections and Unions

Some objects are defined by the intersections and unions of simpler loci. Convex polytopes, for example, are defined by bounding linear pieces. We simply note that the resulting boundaries and joints may be thought of as singularities and understood as in the previous subsection. Note that usually these singularities are cleanly nested by dimensionality, and may possibly be teased apart into a partition *a priori*. Despite the problems of seeing all levels of singularities, the first example in the previous Examples section shows the usefulness of applying section and projection to a structure made up of quite a few jointed flat pieces.

4.3 Quantization

Many objects of interest are not piecewise continuous, but are made up of collections of isolated points. This is the typical case in statistics, where empirical multivariate distributions are made up of a set of observed data points. It is also typically true for computer renditions of continuous mathematical objects: the object is approximated by quantized sample. The point-cloud composition of such loci

is no problem for projection, since it will preserve the image of points. (The projection of a point-cloud is still a point-cloud.) It is a problem for section, however, since a random section will *almost surely* miss all points in a finitely dense sample. There are several possible solutions that may work in various circumstances.

The first approach is to select sections carefully, so as to insure that they go through points. This might be possible, for example if the locus is generated by sampling in a regular grid. This was the solution used for the example loci of the previous section. Caution is needed, however, since some such convenient sections may be singular (i.e., w.r.t. the *almost surely* conditions). Also new aliasing artifacts may be introduced, since the section operation will amount to a yet-more-sparsely sampled version of the true section image.

A second solution available in some mathematical cases, is to generate the section loci explicitly. That is, instead of generating a sampled version of the full object and trying to section it, it may be possible to first specify the parameters of a sectioning hyperplane, and then explicitly generate a version of the object exactly as it intersects that hyperplane.

A third solution is to try to "smooth" the locus in some sense, that is by interpolating between points in some local region to make a continuous approximation which can be treated directly.

A fourth solution is to make "fat" points, i.e. make the points in the cloud spheres of finite radius, so there is a finite chance of hitting them with a section.

A final solution involves taking "thick" sections, i.e., ones with finite volume so that they can intersect some of the points. A "thick" section would capture all points in the locus within some distance, δ, of the sectioning hyperplane. This would be accomplished by intersecting the locus with a generalized cylinder, the Cartesian product of a m-flat and a $(n-m)$-sphere of radius δ, and projecting the intersection set onto the $m-flat$. It is the final projection operation that maintains the visibility of the points.

The success of any of these methods depends on *scale*: the scale of quantization must be sufficiently small w.r.t. scale of meaningful structure. This will help prevent aliasing problems in all the methods. It keeps the notion of neighbors simple for smoothing. With thick sections, it is what may make it possible to find a thickness, δ, that is sufficiently larger than scale of quantization that slices will not usually miss points, yet smaller than scale of structure so that the thickness will not blur the global structure. Of course if the scales of quantization and structure are too close together, then there are intrinsic limits on the adequacy of the rendition in the full space. How much more latitude is needed for section and projection is not yet clear.

4.4 Noise

A final deviation from the idealization is noise. Empirical statistical loci typically have the structure of interest obscured by noise, i.e., random perturbations of the positions of the points. Again scale seems the key: If the scale of the noise is small with respect to that of meaningful structure, then there should be no serious problems. If the scale of noise gets too large, then the structure's image under section and projection may be obscured. But in such cases, one might argue that the "true" shape of the locus has become problematic in a more theoretically fundamental way.

5. Discussion

This paper has examined some formal capabilities of the two geometric transformations section and projection. It was shown that they have complementary strengths and weakness in revealing structure of various dimensionality, and that together they form a powerful composition.

Although the systematic joint use of section and projection should help the investigation of high dimensional loci, a number of difficulties remain. The challenges presented by singularities of codimension, $p<m-1$, and quantization effects have already been mentioned. By far the most important outstanding problem regards the comprehensive assessment of shape. Section and projection are certainly among the fundamental graphical tools for getting relevant information, but the geometry of higher dimensions is fantastically rich, and even the most informative individual 2-D images can only capture glimpses of aspects of the shape.

Thus there are at least three important major directions of future research. The first involves getting the most from each low dimensional image, which requires understanding what these transformations do to a variety of features of a locus. The feature of dimensionality was the focus of this paper. Examples of other important aspects (with some conjectured results given in parentheses) are: what do the transformations do to simple aspects like distances (projection shortens but never lengthens them; section preserves them), angle, position, and orientation; convexity (preserved in both), polytopality (preserved by both -- but what about the number of faces, etc.), connectedness (preserved in projection, but not section). A systematic understanding of these will enrich the ability to understand how a given picture relates to the object pictured.

A second direction for future work is how to make use of other techniques, such as projections that preserve density information, *probing* (a technique closely related to projection), the use of regular sampling grids, etc.

The third major direction involves the efficient collection and assembly of multiple glimpses to capture the whole structure. There has been considerable work on algorithms for the assessment of shape from projections, motivated by the field of tomography [19] [20] [21] There has also been some general work on inferring shapes of polytopes using probing[22]. Further work, encompassing both section and section-then-projection will be needed.

An additional, independent issue, concerns the psychological aspects of high-dimensional visualization. The formal treatments can explore the question about what kind of information is theoretically available from various tools, information that could be used by some arbitrary intelligent machine. It is a further question what kind of information can be captured and integrated by human intelligence, for example to support useful valid inference about the locus as a result of the low dimensional views.

REFERENCES

1. Hoffman, David (1987) "The computer-aided discovery of new embedded minimal surfaces," *The Mathematical Intelligencer* 9(3), 8-21.

2. Banchoff, Thomas F. (1980) "Computer animation and the geometry of surfaces in 3-and 4-space," *Proceedings of the International Congress of Mathematicians. Helsinki, 1978.* Acad. Sci. Fennica: Helsinki, 1005-1013.

3. Banchoff, Thomas F. (1982) "Computer graphics in geometric research," *Recent trends in mathematics, Reinhardsbrunn 1982* Tubner: Leipzig, 316-327.

4. Tukey, J.W., Friedman, J.H., and Fisherkeller, M.A. (1976) "PRIM-9, an interactive multidimensional data display and analysis system." In *Proceedings of the 4th International COngress for Stereology,* Sept 4-9, 1975, Giathersburg, Maryland.

5. Donoho, D.L., Huber, C., Ramos, E., and Thoma, M. (1982) "Kinematic Display of Multivariate Data," *Proceedings of the Third Annual Conference and Exposition of the National Computer Graphics Association.*

6. Friedman, J.H., McDonald, J.A., and Stuetzle,W. (1982) "An Inroduction to Real Time Graphics for Analyzing Multevariate Data," *Proceedings of the Third Annual Conference and Exposition of the National Computer Graphics Association*

7. McDonald, J.A. (1982) *Interactive Graphics for Data Analysis,* Ph.D. Thesis, Stanford University.

8. Tukey, P.A. and Tukey, J.W. (1982) "Some graphics for studying four-dimensional data." *Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface,* Heiner, K.W., Sacher, R.S., and Wilkinson, J.W., (eds.) New York: Springer-Verlag, 60-66.

9. Tukey, P.A. and Tukey, J.W. (1981) "Graphical display of data sets in 3 or more dimensions," Chapters 10,11, and 12 in *Interpreting Multivariate Data* V. Barnett (ed.) Wiley, London.

10. Nicholson, W.L., and Carr, E.B. (1985) "Looking at more than three dimensions," *Computer Science and Statistics: The Interface* L. Billard (ed.) North-Holland, 201-209.

11. Asimov, D. (1985) "The grand tour: A tool for viewing multidimensional data" *SIAM Journal on Scientific and Statistical Computing,* 6(1), p. 128-143.

12. Buja, A., Asimov, D., Hurley C. and McDonald J. A. (1988) "Elements of a Viewing Pipeline for Data Analysis," in W. S. Cleveland (Ed.), a book to appear on high-interaction graphics, Wadsworth, Statistics/Probability Series.

13. Chernoff, H.(1973) "The use of faces to represent point in k-dimensional space graphically," *Journal of the American Statistical Association, 68,* 361-368

14. Kleiner, B. and Hartigan, J.A. (1981) "Representing points in many dimensions by trees and castles," *Journal of the American Statistical Association, 76* (374), 260-269.

15. Furnas, George W. (1988) "Metric Family Portraits," *Journal of Classification, 5 (2),* (in press).

16. Nicholson, W. L. and R.J. Littlefield (1982) "The use of color and motion to display higher-dimensional data," in *Proceedings of the Third Annual Conference and Exposition of the National Computer Graphics Association, Inc.* 1:476-485.

17. Furnas, G. (1988) "Dimensionality constraints on projection and section views of high dimensional loci", *Bellcore Technical Memorandum.*

18. Friedman, J.H., and Tukey, J.W. (1974) "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comp. C-23* 881-890.

19. Shepp, L.A. Kruskal, J.B. (1978) "Computerized Tomography: The New Medical X-Ray Technology" *Amer. Math. Monthly, 85* (6), 420-439

20. Reconstructing a set or measure from finitely many parallel or central projections, J. H. B. Kemperman' [Talk at given 3/17/88 at Bellcore, Morristown N.J.]

22. Dobkin, D.P., Edelbrunner H. and Yap, C.K. (1986) "Probing convex polytopes," *Proc. 18th Ann. ACM Symposium on Theoretical Computing,* 424-432.

A DEMONSTRATION OF THE DATA VIEWER

Catherine Hurley
University of Waterloo

ABSTRACT

We have designed and implemented a program called data viewer for exploring multivariate data sets. The program produces plots moving in real-time by projecting onto a sequence of user-controlled planes. Multiple plots may be simultaneously controlled, allowing dynamic comparisons of data sets. In this presentation, we demonstrate the data viewer by describing and interpreting a selection of plots.

1. Introduction

Recent computing advances have encouraged the development of new data analytic methods, many of them graphical in nature (Cleveland 1987). We have been concerned with graphical methods for analyzing multivariate data. Typically, multivariate data is projected onto some low (one or two) dimensional subspace prior to display. Motion graphics present us with one way of improving on the resulting display -- simply show a new projection every fraction of a second. The PRIM system of Fisherkeller, Friedman and Tukey (1974) was an early demonstration of this technique; they used motion to display a rotating 3-d point cloud. Programs for 3-d rotations have become widely available in the last few years. In our data viewer program, we go beyond 3-d rotations and use motion to display data sets with arbitrary numbers of variables. Briefly, the program produces moving plots by projecting the observations onto smoothly changing sequences of planes. This presentation demonstrates the data viewer by describing and interpreting a selection of plots.

As background, we mention some important aspects of the data viewer design. These will be illustrated throughout the sections which follow.

- **Constructing moving projections**
  We consider 2-d projections displayed by a scatterplot, and 1-d projections displayed with a marginal density estimate. Changing the projection results in a moving scatterplot or density plot appearing on the screen. With the data viewer, the user controls the sequence of projections, implying also that he/she may choose particular projections for display. The projection sequence is constructed by interpolating between consecutive elements of a user-chosen sequence of target planes. For more details, see Hurley (1987), Hurley and Buja (1988).

- **The user-interface**
  Real-time, rather than animated, motion is preferable for analyzing data. All data viewer plots are produced in real-time, which calls for real-time user-controls. For this reason, we equip the program with a graphical user-interface, where the user communicates with the program by pointing a mouse at some part of the data viewer display, and depressing a mouse button. Further details are given in Buja et al (1987), Hurley (1987).

2. The data viewer window

The data viewer program produces plots in some area on the screen which we refer to as a data viewer window. Figure 1 shows one such window, displaying a view of the `St.Helens` data set. This data set contains 680 observations on earthquakes occurring in the vicinity of Mount St. Helens, during May, 1980, where the quantities recorded are date, latitude, long-

itude, depth and magnitude.

There are a fixed set of items appearing in a data viewer window. These are a plot, a title, the variable boxes on the left hand side, a control panel in the lower left corner, and a plot interaction menu lying next to the control panel. Each of these items displays some information relevant to the user. In addition, the items are mouse sensitive, and respond to mouse clicks by changing their appearance. Most user-program interaction occurs in this way. For example, by clicking on various parts of the control panel, the user controls some aspects of the scatterplot motion, such as the speed, and direction (forwards or backwards).

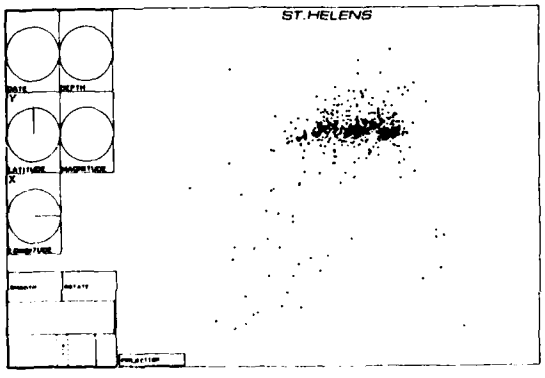


Figure 1: A data viewer window

In figure 1, the boxes for **date** and **latitude** have horizontal and vertical lines drawn from their centers, telling us that the displayed plot is a bivariate scatterplot of **date** and **latitude**. A variable box has a label **X**, **Y**, **A** or blank appearing on the top left hand corner. These labels have a special purpose -- they determine which projections may be shown. An **A** label signifies that the variable is active and may appear in the current projection. With an **X** (**Y**) label, the variable is allowed to have a projection coefficient for the horizontal (vertical) direction only. No label indicates that the variable is inactive, and so has zero horizontal and vertical coefficients. Mouse clicks in the variable boxes are used to change the labels.

The plot interaction menu controls the style of plot interaction, where the current possibilities are point identification, shifting and scaling of the plot axes (see Buja et al, 1987) rotation of the plot in the plane, and moving projections. For instance, when point identification is the current selection, clicking near one of the point symbols causes a label to appear. In the examples presented here, we are concerned with moving projections, so the plot interaction menu shows "PROJECTION". This implies that clicking in the plot region causes a moving projection to appear.

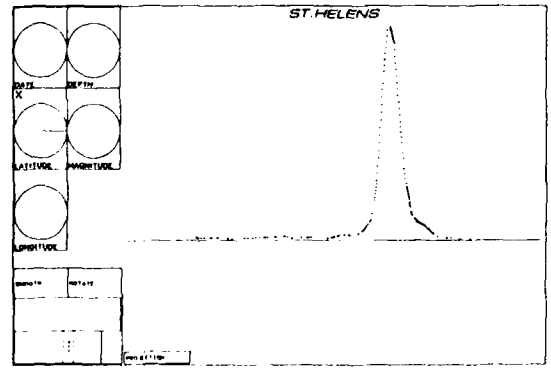


Figure 2: A density estimate

The data viewer program also displays 1-d projections, by plotting a marginal density estimate for the projected observations. For example, figure 2 shows a *density estimate of* **latitude**. (The density estimate is an average shifted histogram (Scott 1985).) As in figure 1, the box for this variable has a horizontal line and an **X** label. Since the plot shows a 1-d projection, there is no box with a vertical line.

3. 3-D Rotations

In figure 3(a), we have picked out the 3-variable subspace consisting of **latitude**, **longitude** and **depth**, by marking their respective boxes with **A** labels. The pair of variables **latitude** and **depth** are in the plane of the screen, while the third, **longitude**, is perpendicular to the screen. Notice that the mouse cursor is positioned on the right hand side of the scatterplot. With a left mouse click at this position, the point cloud rotates towards the mouse cursor. More precisely, the point cloud spins in the direction given by the center of the plot region and the cursor position. A mouse click in the plot region as the points are moving stops the motion. The next click restarts the rotation, in the direction specified by the current position of the mouse cursor. With these controls, the user can spin a 3-d point cloud in any direction. Figure 3(b) shows a picture of the data viewer window after some point cloud rotations.

Notice now that lines are drawn in all three `latitude`, `longitude` and `depth` boxes. The lines are in fact the projections of the three coordinate axes.

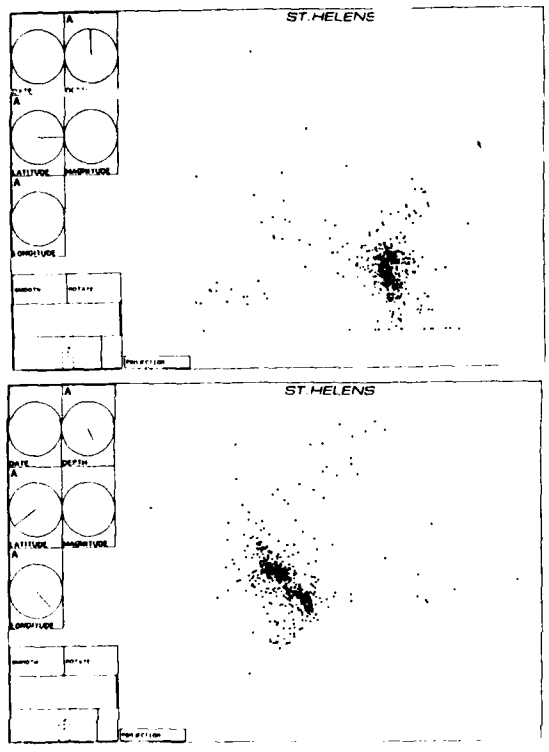


Figure 3: 3-D rotations

4. Linking for close-up views

All of the plots shown so far demonstrate that earthquake locations are highly concentrated, so that it is hard to see the structure of the dense cluster. For this, separate plots of the high-density region are necessary. Suppose the data set `St.Helens-dense` contains the subset of cases in the high-density region. To view this subset separately, we may construct a second data viewer window. Figure 4 shows two data viewer windows, one each for the `St.Helens` and `St.Helens-dense` data sets. In both windows, the cases belonging to the dense subset are drawn with square glyphs, while the remaining points have hollow circular glyphs. By comparing the lines drawn in the variable boxes, we see that the two windows show the same projection. This implies that the scatterplot in the lower window is a "close-up" of the upper scatterplot.

As before, pointing the mouse cursor at the plot region in the upper window and clicking causes the point cloud to rotate. However, this time the point cloud in the lower window also rotates, and in the same direction. This is because the second window was constructed in a special way, in order to *link* it to the existing window. In this case, simultaneous motion of the two scatterplots permits a dynamic data set comparison, because the second window displays throughout a close-up of the first.

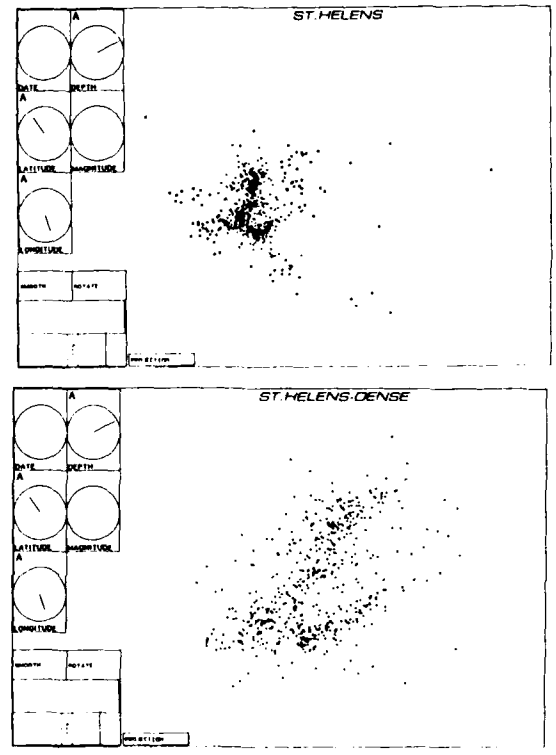


Figure 4: A close-up view

5. Connecting plots

Figure 5 shows a data viewer window for the `Places` data set. This data consists of scores for 329 US cities on 9 criteria, chosen to measure "livability" of the cities (Rand McNally 1986). The nine criteria are climate, housing, health care, crime, transportation, education, the arts, recreation and economics. For housing and crime, the lower the score the better. For all other variables, the higher the score, the better. Three additional variables are included, namely, population (transformed to a log scale), latitude and longitude for each of the 329 cities.

The upper plot, figure 5(a) gives a bivariate scatterplot of latitude and longitude. The two "extra" points on the left hand side of the map represent Anchorage, Alaska, and Honolulu, Hawaii. Their latitude and longitude coordinates have been adjusted so

that all cities fit nicely into the plot region. The middle plot shows a bivariate scatterplot of `climate` and `housing`. Instead of changing the display immediately from one bivariate scatterplot to another, we can gain a lot of information by watching a smooth progression from one scatterplot to another, and back again. We call this *connecting* the scatterplots. In this way, we discover which U.S. cities have good or bad climate, and expensive or cheap housing prices.

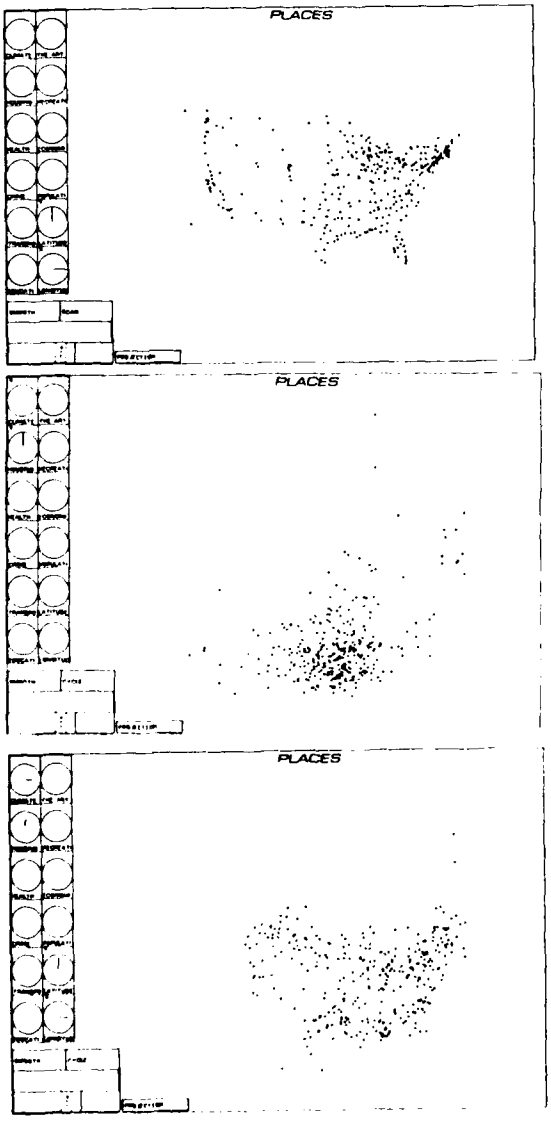


Figure 5: Connecting scatterplots

We construct the sequence of projections which connect the scatterplots shown in figure 5(a) and (b) as follows: Suppose the window currently displays `climate` and `housing`, and we pick `longitude` and `latitude` as the target plot. Motion resumes with a click on a mouse button, proceeding from the current to the target plot. Briefly, the horizontal projection vector rotates in the `climate, longitude` plane, while the vertical projection vector rotates simultaneously at the same rate in the `housing, latitude` plane. When the projection reaches the target, motion pauses momentarily, and then resumes back towards the `climate, housing` plot. The displayed projection continues to cycle between these two scatterplots until the user intervenes.

The third plot in figure 5 shows one of the intermediate projections. From the variable boxes we see that both `climate` and `longitude` have non-zero projections in the horizontal direction, similarly, `housing` and `latitude` in the vertical direction. By watching the smooth progression repeatedly between the pair of scatterplots shown in figure 5 (a) and (b), we gain the following information:

- The cluster of points with the best climate are all Californian cities. They also have high housing prices.
- Highest housing costs are in the vicinity of New York. (The two points with very high scores on `housing` are actually Connecticut cities).
- The mid-west has the worst climate: Minnesota, Wisconsin, and the Dakotas.

6. Linking to compare transformed data

Some of the ratings, in particular `the-arts` and `health-care`, give extremely high scores to the biggest cities-- New York, Chicago and L.A.. This results in scatterplots where most of the observations are clustered together, so that associations between variables are hard to pick out. For this reason, ratings are transformed to normal scores.

Figure 6 shows two data viewer windows, the upper one with the rating variables as before, and the lower one with the normal scores. Both windows display a density estimate for a linear combination of the rating variables. The linear combination is the same since the data viewer windows are linked by common projections. Notice the dot on the extreme right in the upper plot; this is New York. In the lower plot, New York lies far closer to the other cities. As the projection vector moves in the space spanned by the X-variables, we see how the transformation to normal

scores affects marginal distributions. The density estimate in the lower window is generally symmetric, and quite often looks "bell-shaped". For the untransformed ratings, the 1-d projections have highly skewed distributions. With a moving x-vector, the density's peak shifts to and fro across the screen.

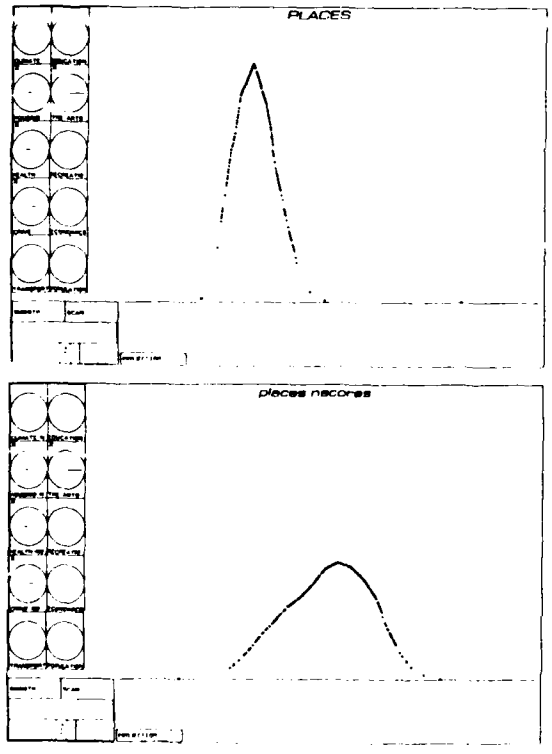


Figure 6: Comparing transformed data

7. *Predictor-response plots*

Most of the nine rating variables tend to assign high values to big cities. To judge the overall nature of the association between population and the ratings, we examine plots of population against linear combinations of the rating variables. Suppose we pick **population** as the single Y-variable, and make each of **the-arts**, **health-care**, **economics**, **education** and **recreation** X-variables. (From the bivariate scatterplots, these five have the strongest individual associations with **population**.) Then, motion yields a plot of **population** against a changing linear combination of the five X-variables.

By watching the moving scatterplot, we discover a projection with high x-y association, as shown in figure 7(a). We can see that **population** is linearly related to a weighted average of the five selected rating variables. Also, **health-care** and **the-arts** have the largest coefficients, whereas the coefficients for **economics** and **recreation** are comparatively small. (The variables have been transformed to normal scores, so that it is reasonable to compare their projection coefficients.)

Do the variables **economics** and **recreation** have a negligible contribution to the x-y association in the above projection? We may answer this question as follows. Suppose we deactivate the two variables **economics** and **recreation**, thus requiring them to have zero projection coefficients in succeeding target planes. In particular, the x-vector for the next target will be the current x-vector orthogonalized with regard to the two deactivated variables. With a rotation towards this target, we receive a visual impression of how the quality of the x-y association deteriorates (if at all), as the coefficients of the two variables shrinks to zero. The second plot in figure 7 shows the projection onto the new target. Overall, it looks very similar to the previous plot, with most changes occurring among cities with lower population. As far as the eye can judge, **economics** and **recreation** do not contribute to the x-y association observed in the upper plot.

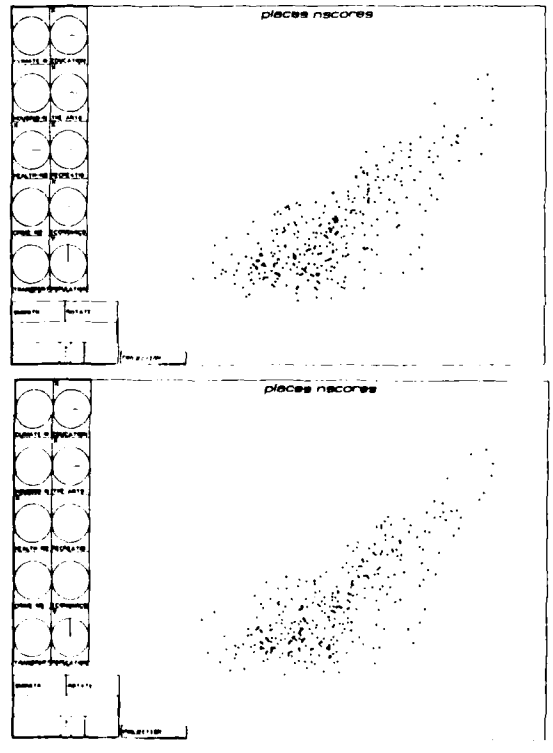


Figure 7: Exploratory regression

8. Data derived variables

The data viewer can also display plots of principal components, canonical variates or the linear discriminants. Indeed, the user may choose any linear combinations to form additional variables, but as a rule, data derived combinations will be the most useful.

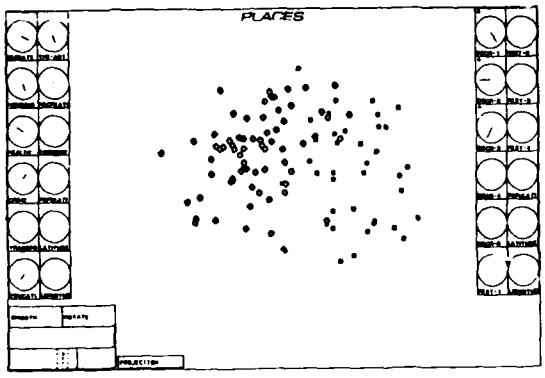


Figure 8: Plotting linear discriminants

Figure 8 shows a data viewer window for the **Places** data, with additional boxes on the right hand side for some new variables. In this case, the new variables were obtained by performing a discriminant analysis using the nine ratings, where the cities where classed by location into (i) west coast states plus Alaska and Hawaii, (ii) Rocky mountain states, (iii) mid-west states, (iv) south-west states, (v) south-east states and (vi) north-east states. The purpose is to discover how the ratings vary across locations.

The user first marks each of the location groups with a different plotting symbol, and then asks the data viewer to compute the discriminants on the basis of the ratings. When the calculations are complete, the data viewer redraws its window, with more boxes on the r.h.s. for the derived variables. The additional boxes are ordered column-wise, and labeled **discr-1** through **discr-5** for the discriminants, **rest-1** ... **rest-4** for some dummy variables, and followed by **population**, **latitude** and **longitude**.

Now the user can specify (moving) projections in terms of either the l.h.s. or r.h.s. variables. Figure 8 displays a projection obtained by performing 3-d rotations in the space spanned by the first three discriminants. This projection gives good separation of the west coast and north-east states in the horizontal direction. For a clearer presentation of the two groups, they are marked with large squares and open circles respectively, while cities in other regions are not shown. The l.h.s. boxes show which rating variables contribute to the separation. Note that

- **climate** has a large positive coefficient in the horizontal direction; since west coast cities lie to the right of east coast cities in the scatterplot, this implies that the west coast has better climate.
- **Health-care** and **education** have moderately sized, but negative, horizontal coefficients. Therefore, it seems as if east coast cities offer superior health-care and education facilities.
- **Recreation**, **transportation** and **economics** have little or no impact on the separation observed.

9. Conclusion

This presentation aimed to illustrate some of the capabilities of the data viewer program, through describing some of the displays produced. With a system which relies so heavily on real-time motion and real-time graphical interaction, a textual description of a few static plots is at best a poor substitute for a "live" demonstration. However, we would hope to have convinced the reader of the potential of data analysis tools such as data viewer.

Acknowledgements

The author wishes to thank A. Buja and J.A. McDonald for helpful discussions.

References

Buja, A., Asimov, D., Hurley, C., McDonald, J. A. (1987) "Elements of a Viewing Pipeline for Data Analysis", in *Dynamic Graphics for Statistics*, eds. W. S. Cleveland and M. E. McGill, Monterey, CA: Wadsworth.

Cleveland, W.S. (1987) "Research in Statistical Graphics", *J. Am. Stat. Assoc.* vol. 82, 398.

Fisherkeller, M. A., Friedman, J. H., Tukey, J. W. (1974) "PRIM-9, An Interactive Multidimensional Data Display and Analysis System", *Proceedings of the Pacific ACM Regional Conference.*

Hurley, C. (1987) "The Data Viewer: A Program for Graphical Data Analysis", Ph.D. thesis and tech. report, Stat. dept., University of Washington.

Hurley, C., Buja, A. (1988) "Analyzing High-dimensional Data with Motion Graphics" tech. report STAT-88-03, Dept. of Stats & Act. Sci., University of Waterloo.

Rand McNally (1986) *Places Rated Almanac*

Scott, D. W. (1985) "Average Shifted Histograms: Effective Non-Parametric Density Estimation in Several Dimensions", *Annals of Statistics* 13, p. 1024-1040.

This research was supported in part by U.S. Dept. of Energy under contract DE-FG06-85ER25006 when the author was at the University of Washington, Seattle.

VISUALIZING MULTI-DIMENSIONAL GEOMETRY WITH PARALLEL COORDINATES

Alfred Inselberg * # & Bernard Dimsdale *

KEY WORDS: Multi-Dimensional Graphics, Multi-Dimensional Visualization, Duality, Parallel Coordinates

* *IBM Scientific Center*
11601 Wilshire Boulevard
Los Angeles, CA 90025-1738
aiisreal@ibm.com

&

Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0782
inselber@oberon.usc.edu

ABSTRACT

By means of parallel coordinates a non-projective mapping between subsets of R^N into subsets of R^2 (i.e. $2^{R^N} \rightarrow 2^{R^2}$) is obtained. In this way not only N-tuples but also relations among N variables, for any positive integer N, can be visualized in terms of their planar images. These planar diagrams have geometrical properties corresponding to some properties of the N-dimensional relation they represent. Starting from a *point* $\leftrightarrow$ *line* duality when $N = 2$, the representation of lines in R^N is given and illustrated by an application to Air Traffic Control (i.e. for R^4). It is followed by the representation of hyperplanes, and more general hypersurfaces. There is an algorithm for constructing and *displaying any interior point* to such a hypersurface showing some *local (i.e. near the point)* properties of the hypersurface and information on the point's *proximity* to the boundary.

Introduction

Other than a superficial similarity to Nomography Parallel Coordinates were first formulated in 1978 with the first report appearing in 1981 (see [12]). They provide a methodology for visualizing not only N-Dimensional points but also N-Dimensional Hypersurfaces (i.e. relations among N variables) for *arbitrary* N the method being the same for every N. Other methodologies (but not suited for multivariate relations) are well known (see [1], [2], [4] and the bibliographies in [5] and [15], for example). Some applications of parallel coordinates can be found in [6], [7], [12], [13], [15], [16], [17] and [18].

On the plane with xy-Cartesian coordinates, and starting on the y-axis, N copies of the real line, labeled $x_1, x_2, \ldots, x_N$, are placed equidistant and perpendicular to the x-axis. They are the axes of the *parallel coordinate* system for Euclidean N-Dimensional Space R^N all having the same positive orientation as the y-axis --see Figure 1. A point C with coordinates $(c_1, c_2, \ldots, c_N)$ is represented by

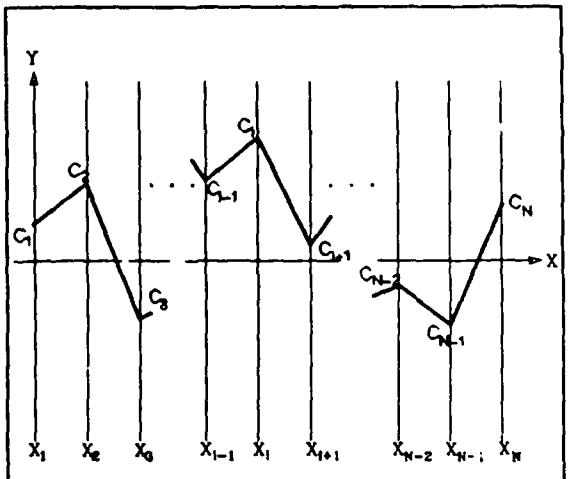


Figure 1: -- Parallel axes for R^N.

The polygonal line shown represents the point $C = (c_1, \ldots, c_{i-1}, c_i, c_{i+1}, \ldots, c_N)$.

the polygonal line whose N vertices are at $(i-1, c_i)$ on the x_i-axis for $i = 1, \ldots, N$. In effect, a 1-1 correspondence between points in R^N and planar polygonal lines with vertices on $x_1, x_2, \ldots, x_N$ is established. A *convex* hypersurface in R^N is represented by the envelope of the family of polygonal lines representing all points on the hypersurface (see [3]). In short, a non-projective mapping $2^{R^N} \rightarrow 2^{R^2}$ is established. The key idea is that the description of a higher dimensional object is captured, to a considerable extent, in the *2-dimensional representation* of the envelope of the polygonal lines representing its points.

Points are denoted by capitals and lines (or arcs of curves) by lower-case letters respectively. In parallel coordinates, the corresponding symbols are shown with a bar superscript (i.e. $\bar{\ell}$ represents the line ℓ, $\bar{P}$ represents the point P etc.).

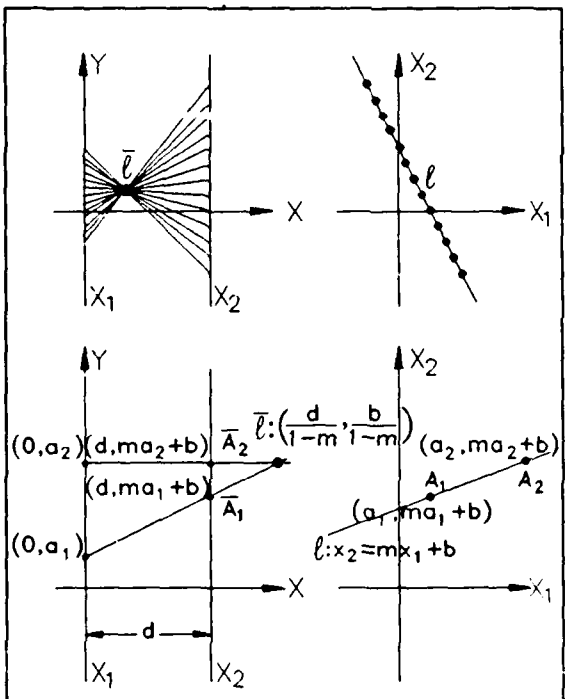


Figure 2: -- In the plane parallel coordinates induce duality.

Top part shows the family of segments, representing points on a given line, *intersecting* at a point. Lower part shows the duality line $\ell \leftrightarrow \bar{\ell}$ point in general.

The Fundamental Point ↔ Line Duality

Points on the plane are represented by segments between the x_1 and x_2-axis and, in fact, by the *line* containing the segment. In Figure 2, the distance between the x_1 and x_2 axes is "d". The line

$$\ell : x_2 = mx_1 + b, \quad m < \infty$$

is the collection of points A. They are represented by the infinite collection of lines $\bar{A}$ on the xy-plane which when $m \neq 1$ intersect at the *point*:

$$\bar{\ell} : \left(\frac{d}{1-m}, \frac{b}{1-m} \right),$$

given with respect to the xy-Cartesian coordinates The reason for representing the point P by the *whole* line $\bar{P}$, rather than just the *segment* between the parallel axes, is that $\bar{\ell}$ may lie outside the strip between the axes. For lines with $m = 1$, we consider xy and x_1x_2 as two copies of the *Projective Plane* [8] so that the line ℓ corresponds to the *ideal point* $\bar{\ell}$ *with tangent direction (i.e. slope)* b/d. Conversely, in the x_1x_2-projective plane the *ideal point* with slope m is mapped into the *vertical line* at $x = d/(1-m)$ of the xy-projective plane. Hence, we have a *duality* between points and lines of the Projective Plane. This duality as expressed by means of homogeneous coordinates is a linear transformation a--*correlation*--between the line coordinates $[m, -1, b]$ of ℓ and the point coordinates $(d, b, 1-m)$ of $\bar{\ell}$:

$$C_A : \ell \leftrightarrow \bar{\ell}, \quad (\bar{\ell}) = A[\ell]$$

where $[\ell]$ and $(\bar{\ell})$, the line and point (homogeneous) coordinates respectively, are taken as *column vectors* and A is a non-singular 3x3 matrix.

By means of the correlation C_A above the collection of points on a curve is mapped into a collection of lines which can be considered as tangents to another curve. On the plane conics map into conics (see [9]). Actually, this property is more general and applies to *generalized conics*. Consider a double cone whose base is *any* bounded convex set as shown in

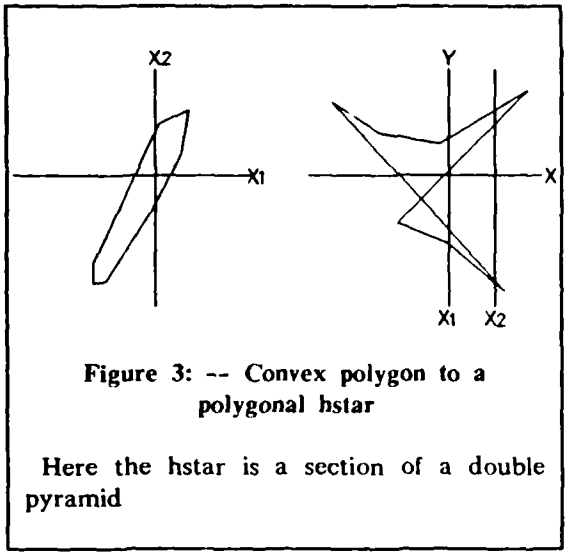


Figure 3: -- Convex polygon to a polygonal hstar

Here the hstar is a section of a double pyramid

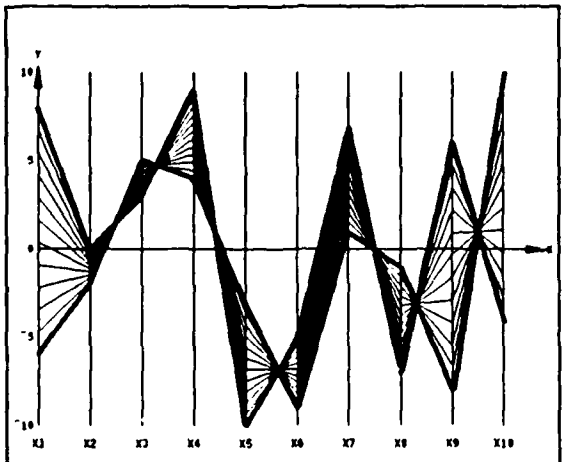


Figure 4: -- Interval on a line in R^{10}.

Figure 5. As in the ordinary conics, three kinds of planar sections exist, those having *bounded, unbounded* or *two disjoint unbounded* components. By analogy to the ordinary conics they are called *estars, pstars and hstars* (the "e" for ellipse, "p" for parabola and "h" for hyperbola) respectively. Collectively, they are referred to as *gconics*. It turns out that gconics map into gconics (see [13]) and in particular estars map into hstars shown in Figure 3. This yields a new duality betweem bounded and unbounded convex

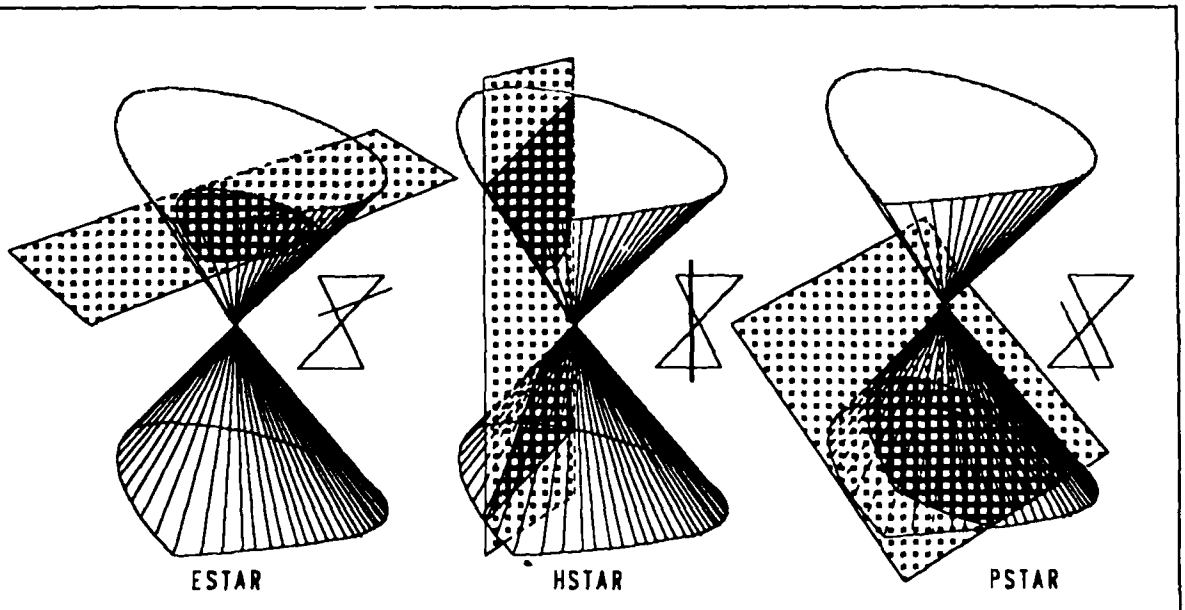


Figure 5: -- Generalized Conics (Gconics)

sets and hstars as well as a duality between Convex Merge (Convex Union) and Intersection. Based on these results efficient new algorithms for Convex Hull construction, and the Convex Merge and Intersection of Convex sets were derived (see [17]). For non-convex curves there is a surprising duality between cusps and inflection points.as shown

Lines in R^N

Consider now a line ℓ in R^N described by:

$$\ell_{i,i+1} : x_{i+1} = m_i x_i + b_i \quad i = 2, \ldots, N. \qquad m_i \neq 0$$

In the $x_i x_{i+1}$-plane the relation labeled $\ell_{i,i+1}$ is a line and by the correlation C_A translated appropriately it is represented by the *point*

$$\bar{\ell}_{i,i+1} : (i - 1 + \frac{1}{1-m_i}, \frac{b_i}{1-m_i}).$$

There are $N-1$ such independent relations in the given set of equations, ergo the line ℓ is represented by the corresponding $N-1$ points. For example, in Figure 4 we see several points on a line interval in R^{10}. It is clear from the diagram how a point can be constructed on the line, for any given initial value of one of the variables. It is also clear how, given the equations or the coordinates of their equivalent points in parallel coordinates, points on the line can be calculated. It also turns out that the minimum distance between two lines is "visible" in parallel coordinates [16] a useful property in problems involving proximity as in Air Traffic Control see Figure 6. The time axis can be thought of as a "clock" and at any given time T, the position of the aircraft is found by selecting the value of T on the T-axis.

Hyperplanes in R^N

Up to this point a very special and useful fact concerning straight lines has not been mentioned. In two dimensions a line in Euclidean space transforms into a point in parallel coordinates. Every line parallel to such a line also transforms into a point in parallel coordinates. The x-coordinate of every such line is the same as the x-coordinate of every other such line, namely $1/(1-m)$. That is to say, the set of parallel lines in Euclidean coordinates transforms into a vertical line in parallel coordinates. In N-dimensions a set of parallel lines transforms into $N-1$ vertical lines. This is the basis for the representation of any hyperplane by $N-1$ vertical lines and a polygonal line representing one of its points. In Figure 7 a planar relation among industrial data was discovered from this observation.

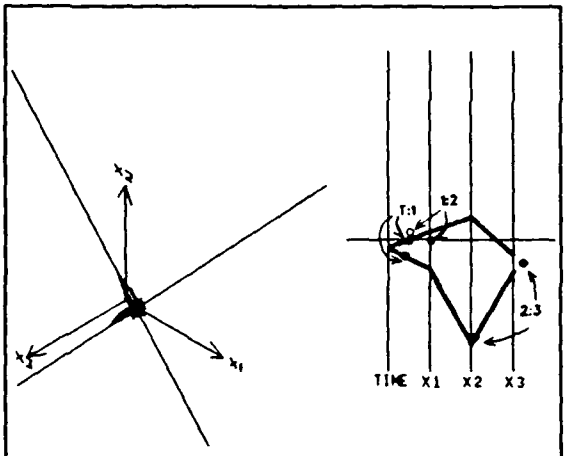


Figure 6: -- The trajectory of an aircraft flying on a straight line path with constant velocity is a line in 4-D and can be represented by 3 stationary points.

On the four parallel axes a polygonal line shows the time, value on the T-axis, when the position (x_1, x_2, x_3) is attained. Even in an accurate 3-D isometric (above left) the aircraft look as if they are almost colliding, the information in parallel coordinates shows that this is not the case.

Hypersurfaces in R^N

A feel for the power of the representation can be gained from Figure 9 from which, with a bit of practice, the vertices, edges and faces, and their interrelationship, of the hypercube can be rec-

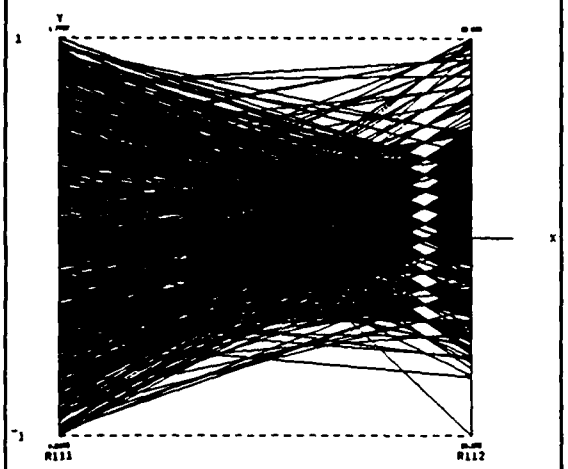


Figure 7: -- Vertical pattern between two variables from a multi-variate set of 2000 industrial data points.

This shows a linear relation between R111, R112 and a third variable.

ognized. The representation of more certain more

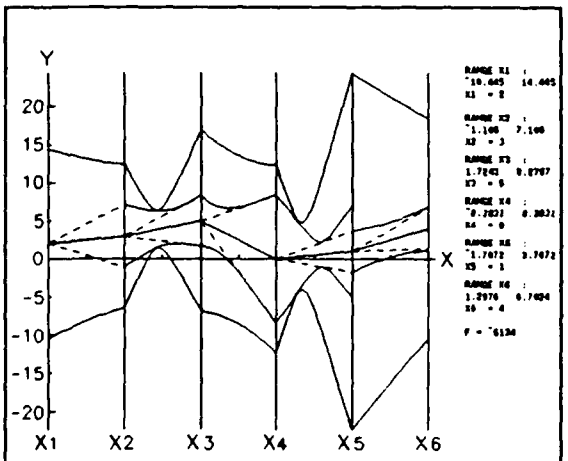


Figure 8: -- The polygonal line represents the point found interior to the Hyperellipsoid

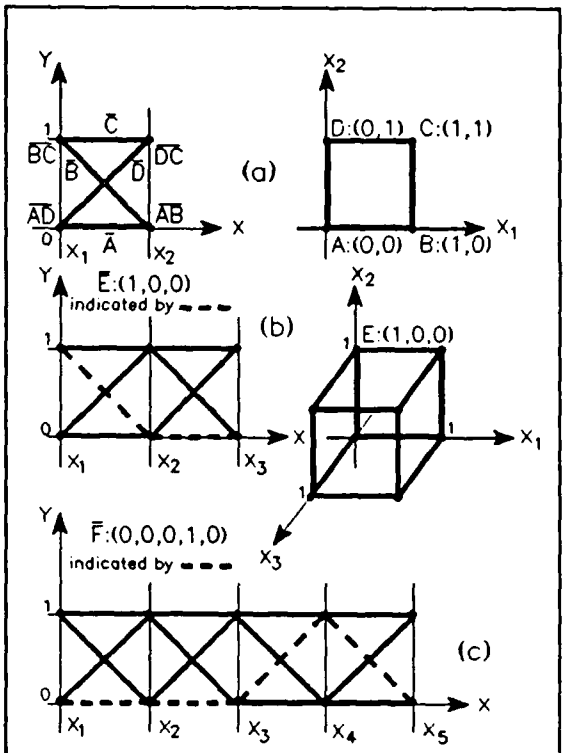


Figure 9: -- Hypercube Representation in Paraliel Coordinates.

Graph of square (a), cube in 3-D (b) and Cube in 5-D (c) all having unit side.

general classes hypersurfaces has been found. There is an algorithm for finding and displaying interior/ exterior or points on the surface as shown in Figure 8.

Though necessarily brief, we hope to have conveyed a notion of a new geometrical tool for visualizing and analyzing multivariate relations. Parallel Coordinates have a "built-in mechanism" for generalizing "lower-dimensional" intuition and results without any intrinsic limit on the dimensionality.

Bibliography

[1] D. Asimov (1985), *The Grand Tour: A Tool For Viewing Multidimensional Data* SIAM J. of Scient. & Stat. Comp. 6:128-143,

[2] V. Barnett Edit. (1981), **Interpreting Multivariate Data**, Wiley, New York,

[3] V. G. Boltyanskii (1964), **Envelopes**, Translated from the Russian by R. B. Brown, Pergamon Press, New York,

[4] D. Brissom Edit. (1979), **Hypergraphics: Visualizing Complex Relationships in Art, Science and Technology** Amer. Assoc. Adv. Sc., Westview Press, Boulder Colorado,

[5] J.M. Chambers, W.S. Cleveland, B. Kleiner & P.A. Tukey (1983) **Graphical Methods for Data Analysis** Duxbury Press, Boston,

[6] T. Chomut (1987), **Exploratory Data Analysis in Parallel Coordinates**, IBM Los Angeles Scientific Center Report # G320-2811,

[7] S. Cohan, D. C. Yang (1986) *Mobility Analysis Of Planar Four-Bar Mechanisms Through the Parallel Coordinate System*, Mech. & Mach. Theor. 21:63-71,

[8] H. S. M. Coxeter (1974), **Projective Geometry**, Univ. of Toronto Press, Toronto,

[9] B. Dimsdale (1981), **Conic Transformations**, *IBM Los Angeles Scientific Center Report* # G320-2713,

[10] B. Dimsdale (1981), **Operating Point Selection for Multivariate Systems**, IBM Los Angeles Scientific Center Report # ZZ20-6249,

[11] B. Dimsdale (1982), The Hyperesp, IBM Los Angeles Scientific Center Report (Unpublished),

[12] A. Inselberg (1981), **N-Dimensional Graphics Part I: Lines & Hyperplanes**, IBM Los Angeles Scientific Center Report # G320-2711,

[13] A. Inselberg (1985), *The Plane with Parallel Coordinates*, Special Issue on *Computational Geometry* The Visual Computer 1:69-91,

[14] A. Inselberg, B. Dimsdale (1986), *Intelligent Process Control & Integrated Instrumentation*, First Conference on Intelligent and Integrated Manufacturing Anaheim, CA, Dec. 1986, ASME vol.21 341-358,

[15] A. Inselberg, B. Dimsdale (1988) *Multidimensional Lines I : Representations*, submitted for publication,

[16] A. Inselberg, B. Dimsdale (1988) *Multidimensional Lines II : Distance and Proximity* , submitted for publication,

[17] A. Inselberg, M. Reif, T. Chomut (1987) *Convexity Algorithms in Parallel Coordinates*, J. of ACM 34 765-801,

[18] E. Wegman (1986) **Hyperdimensional Data Analysis Using Parallel Coordinates**, Tech. Rep. #1, Center for Computational Statistics and Probability, George Mason Univ., Fairfax, VA,

On Some Graphical Representations of Multivariate Data

Masood Bolorforoush and Edward J. Wegman
George Mason University

1. *Introduction.* The classic scatter diagram is a fundamental tool in the construction of a model for data. It allows the eye to detect such structures in data as linear or nonlinear features, clustering, outliers and the like. Unfortunately, scatter diagrams do not generalize readily beyond three dimensions. For this reason, the problem of visually representing multivariate data is a difficult, largely unsolved one. The principal difficulty, of course, is the fact that while a data vector may be arbitrarily high dimensional, say n, Cartesian scatter plots may only easily be done in two dimensions and, with computer graphics and more effort, in three dimensions. Alternative multidimensional representations have been proposed by several authors including Chernoff (1973), Fienberg (1979), Cleveland and McGill (1984a) and Carr et al. (1986).

An important technique based on the use of motion is the computer-based kinematic display yielding the illusion of three dimensional scatter diagrams. This technique was pioneered by Friedman and Tukey (1973) and is now available in commercial software packages (Donohoe's MacSpin and Velleman's Data Desk). Coupled with easy data manipulation, the kinematic display techniques have spawned the exploitation of such methods as projection pursuit (Friedman and Tukey, 1974) and the grand tour (Asimov, 1985). Clearly, projection-based techniques lead to important insights concerning data. Nonetheless, one must be cautious in making inferences about high dimensional data structures based on projection methods alone. It would be highly desireable to have a simultaneous representation of all coordinates of a data vector especially if the representation treated all components in a similar manner. The cause of the failure of the standard Cartesian coordinate representation is the requirement for orthogonal coordinate axes. In a 3-dimensional world, it is difficult to represent more than three orthogonal coordinate axes. We propose to give up the orthogonality requirement and replace the standard Cartesian axes with a set of n parallel axes.

2. *Parallel Coordinates.* We propose as a multivariate data analysis tool the following representation. In place of a scheme trying to preserve orthogonality of the n-dimensional coordinate axes, draw them as parallel. A vector $(x_1, x_2, \ldots, x_n)$ is plotted by plotting x_1 on axis 1, x_2 on axis 2 and so on through x_n on axis n. The points plotted in this manner are joined by a broken line. Figure 2.1 illustrates two points (one solid, one dashed) plotted in parallel coordinate representation. In this illustration, the two points agree in the fourth coordinate. The principal advantage of this plotting device is clear. Each vector $(x_1, x_2, \ldots, x_n)$ is represented in a planar diagram so that each vector component has essentially the same representation.

The parallel coordinates proposal has its roots in a number of sources. Griffen (1958) considers a 2-dimensional parallel coordinate type device as a method for graphically computing the Kendall tau correlation coefficient. Hartigan (1975) describes the "profiles algorithm" which he describes as "histograms on each variable connected between variables by identifying cases." Although he does not recommend drawing all profiles, a profile diagram with all profiles plotted is a parallel coordinate plot. There is however far more mathematical structure, particularly high dimensional structure, to the parallel coordinate diagram than Hartigan

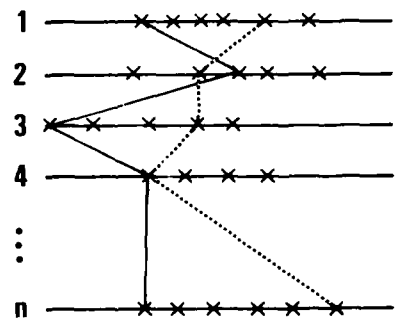


Figure 2.1 Parallel coordinate representation of two n-dimensional points.

exploits. Inselberg (1985) originated the parallel coordinate representation as a device for computational geometry. His 1985 paper is the culmination of a series of technical reports dating from 1981. Finally we note that Diaconis and Friedman (1983) discuss the so-called M and N plots. Their special case of a 1 and 1 plot is a parallel coordinate plot in two dimensions. Indeed, the 1 and 1 plot is sometimes called a before-and-after plot and has a much older history. The fundamental theme of this paper is that the transformation from Cartesian coordinates to parallel coordinates is a highly structured mathematical transformation, hence, maps mathematical objects into mathematical objects. Certain of these can be given highly useful statistical interpretations so that this representation becomes a highly useful data analysis tool.

3. *Parallel Coordinate Geometry.* The parallel coordinate representation enjoys some elegant duality properties with the usual Cartesian orthogonal coordinate representation. Consider a line $\mathcal{L}$ in the Cartesian coordinate plane given by $\mathcal{L}$: $y=mx+b$ and consider two points lying on that line, say (a, ma+b) and (c, mc+b). For simplicity of computation we consider the xy Cartesian axes mapped into the xy parallel axes as described in Figure 3.1. We superimpose a Cartesian coordinate axes t,u on the xy parallel axes so that the y parallel axis has the equation $u=1$. The point (a, ma+b) in the xy Cartesian system maps into the line joining (a, 0) to (ma+b, 1) in the tu coordinate axes. Similarly, (c, mc+b) maps into the line joining (c, 0) to (mc+b, 1). It is a straightforward computation to show that these two lines intersect at a point (in the tu plane) given by $\bar{\mathcal{L}}$: $(\, b(1-m)^{-1}, (1-m)^{-1})$. Notice that this point in the parallel coordinate plot depends only on m and b the parameters of the original line in the Cartesian plot. Thus $\bar{\mathcal{L}}$ is the dual of $\mathcal{L}$ and we have the interesting duality result that points in Cartesian coordinates map into lines in parallel coordinates while lines in Cartesian coordinates map into points in parallel coordinates.

For $0 < (1-m)^{-1} < 1$, m is negative and the intersection occurs between the parallel coordinate axes. For $m=-1$, the intersection is exactly midway. A ready statistical interpretation can be given. For highly negatively correlated pairs, the dual line segments in parallel coordinates will tend to cross near a single point between the two parallel coordinate axes. The scale of one of the variables may be transformed in

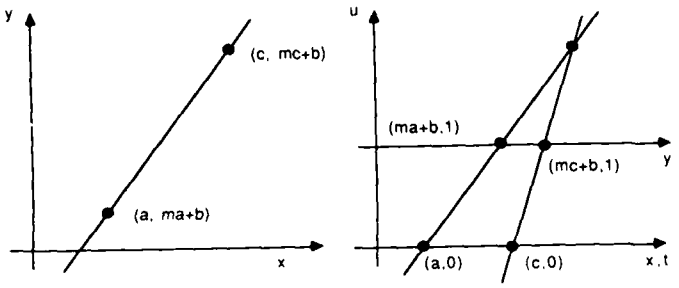


Figure 3.1 Cartesian and parallel coordinate plots of two points. The tu Cartesian coordinate system is superimposed on the xy parallel coordinate system.

such a way that the intersection occurs midway between the two parallel coordinate axes in which case the slope of the linear relationship is negative one.

In the case that $(1-m)^{-1}<0$ or $(1-m)^{-1}>1$, m is positive and the intersection occurs external to the region between the two parallel axes. In the special case $m=1$, this formulation breaks down. However, it is clear that the point pairs are (a, a+b) and (c, c+b). The dual lines to these points are the lines in parallel coordinate space with slope b^{-1} and intercepts $-ab^{-1}$ and $-cb^{-1}$ respectively. Thus the duals of these lines in parallel coordinate space are parallel lines with slope b^{-1}. We thus append the ideal points to the parallel coordinate plane to obtain a projective plane. These parallel lines intersect at the ideal point in direction b^{-1}. In the statistical setting, we have the following interpretation. For highly positively correlated data, we will tend to have lines not intersecting between the parallel coordinate axes. By suitable linear rescaling of one of the variables, the lines may be made approximately parallel in direction with slope b^{-1}. In this case the slope of the linear relationship between the rescaled variables is one. See Figures 3.2 for an illustration of large positive and large negative correlations. Of course, nonlinear relationships will not respond to simple linear rescaling. However, by suitable nonlinear transformations, it should be possible to transform to linearity. The point-line, line-point duality seen in the transformation from Cartesian to parallel coordinates extends to conic sections. An instructive computation involves computing in the parallel coordinate space the image of an ellipse which turns out to be a general hyperbolic form. For purposes of conserving space we do not provide the details here.

It should be noted, however, that the solution to this computation is not a locus of points, but a locus of lines, a line conic. The envelope of this line conic is a point conic. In the case of this computation, the point conic in the original Cartesian coordinate plane is an ellipse, the image in the parallel coordinate plane is as we have just seen a line hyperbola with a point hyperbola as envelope. Indeed, it is true that a conic will always map into a conic and, in particular, an ellipse will always map into a hyperbola. The converse is not true. Depending on the details, a hyperbola may map into an ellipse, a parabola or another hyperbola. A fuller discussion of projective transformations of conics is given by Dimsdale (1984). Inselberg (1985) generalizes this notion into parallel coordinates resulting in what he calls hstars.

We mentioned the duality between points and lines and conics and conics. It is worthwhile to point out two other nice dualities. Rotations in Cartesian coordinates become translations in parallel coordinates and vice versa. Perhaps more interesting from a statistical point of view is that points of inflection in Cartesian space become cusps in parallel coordinate space and vice versa. Thus the relatively hard-to-detect inflection point property of a function becomes the notably more easy to detect cusp in the parallel coordinate representation. Inselberg (1985) discusses these properties in detail.

4. *Further Statistical Interpretations.* Since ellipses map into hyperbolas, we can have an easy template for diagnosing uncorrelated data pairs. Consider Figure 3.2. With a completely uncorrelated data set, we would expect the 2-dimensional scatter diagram to fill substantially a circumscribing circle. As illustrated in Figure 3.2, the parallel coordinate plot would approximate a figure with a hyperbolic envelope. As the correlation approaches negative one, the hyperbolic envelope would deepen so that in the limit we would have a pencil of lines, what we like to call the cross-over effect. As the correlation approaches positive one, the hyperbolic envelope would widen with fewer and fewer cross-overs so that in the limit we would have parallel lines. Thus correlation structure can be diagnosed from the parallel coordinate plot. As noted earlier, Griffen (1958) used this as a graphical device for computing the Kendall tau.

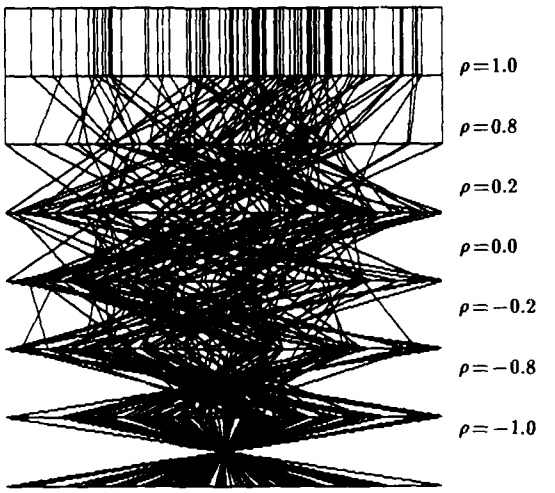


Figure 3.2 Parallel coordinate plot of 6 dimensional data illustrating correlations of $\rho = 1, .8, .2, 0, -.2, -.8$ and -1.

Griffen, in fact, attributes the graphical device to Holmes (1928) which predates Kendall's discussion. The computational formula is

$$r = 1 - \frac{4X}{n(n-1)}$$

where X is the number of intersections resulting by connecting the two rankings of each member by lines, one ranking having been put in natural order. While the original formulation was framed in terms of ranks for both x and y axes, it is clear that the number of crossings is invariant to any monotone increasing transformation of either x or y, the ranks being one such transformation. Because of this scale invariance, one would expect rank-based statistics to have an intimate relationship to parallel coordinates.

It is clear that if there is a perfect positive linear relationship with no crossings, then $X = 0$ and $r = 1$. Similarly, if there is a perfect negative linear relationship, Figure 3.2 is again appropriate and we have a pencil of lines.

Since every line meets every other line, the number of intersections is $\binom{n}{2}$ so that

$$r = 1 - \frac{4\binom{n}{2}}{n(n-1)} = -1.$$

It should be further noted that clustering is easily diagnosed using the parallel coordinate representation.

So far we have focused primarily on pairwise parallel coordinate relationships. The idea however is that we can, so to speak, stack these diagrams and represent all n dimensions simultaneously. Figure 4.1 thus illustrates 6-dimensional Gaussian uncorrelated data plotted in parallel coordinates. A 6-dimensional ellipsoid would have a similar general shape but with hyperbolas of different depths. This data is deep ocean acoustic noise and is illustrative of what might be expected.

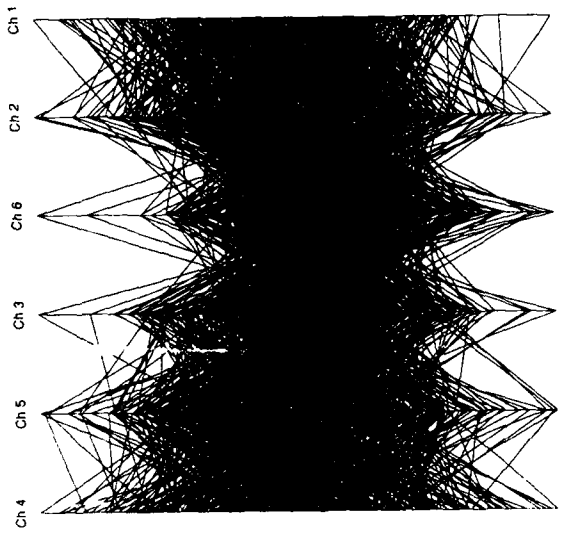


Figure 4.1 Parallel coordinate plot of 6 channel sonar data. The data is uncorrelated Gaussian noise. The second coordinate represents a relatively remote hydrophone and has a somewhat different mean. Notice the approximate hyperbolic shape.

Figure 4.2 is illustrative of some data structures one might see in a five-dimensional data set. First it should be noted that the plots along any given axis represent dot diagrams (a refinement of the histograms of Hartigan), hence convey graphically the one-dimensional marginal distributions. In this illustration, the first axis is meant to have an approximately normal distribution shape while axis two the shape of the negative of a χ^2. As discussed above, the pairwise comparisons can be made. Figure 4.2 illustrates a number of instances of linear (both negative and positve), nonlinear and clustering situations. Indeed, it is clear that there is a 3-dimensional cluster along coordinates 3, 4 and 4.

Consider also the appearance of a mode in parallel coordinates. The mode is, intuitively speaking, the location of the most intense concentration of probability. Hence, in a sampling situation it will be the location of the most intense concentration of observations. Since observations are represented by broken line segments, the mode in parallel coordinates will be represented by the most intense bundle of broken line paths in the parallel coordinate diagram. Roughly speaking, we should look for the most intense flow through the diagram. In Figure 4.2, such a flow begins near the center of coordinate axis one and finishes on the left-hand side of axis five.

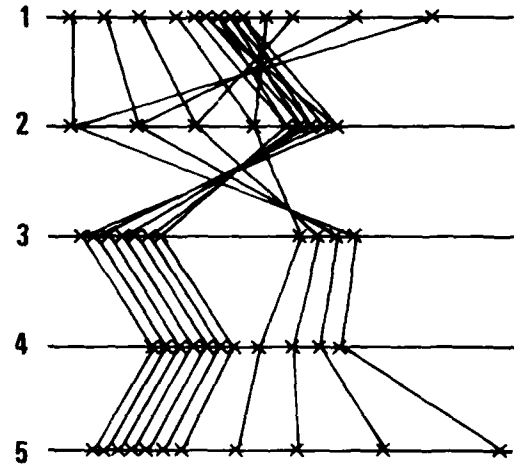


Figure 4.2 A five dimensional scatter diagram in parallel coordinates illustrating marginal densities, correlations, three dimensional clustering and a five dimensional mode.

Figure 4.2 thus illustrates some data analysis features of the parallel coordinate representation including the ability to diagnose one-dimensional features (marginal densities), two-dimensional features (correlations and nonlinear structures), three-dimensional features (clustering) and a five-dimensional feature (the mode). In the next section of this paper we consider a real data set which will be illustrative of some additional capabilities.

5. *An Auto Data Example.* We illustrate parallel cooordinates as an exploratory analysis tool on data about 86 1980 model year automobiles. They consist of price, miles per gallon, gear ratio, weight and cubic inch displacement. For n = 5, 3 presentations are needed to present all pairwise permutations. Figures 5.1, 5.2 and 5.3 are these three presentations. In Figure 5.1, perhaps the most striking feature is the cross-over effect evident in the relationship between gear ratio and weight. This suggests a negative correlation. Indeed, this is reasonable since a heavy car would tend to have a large

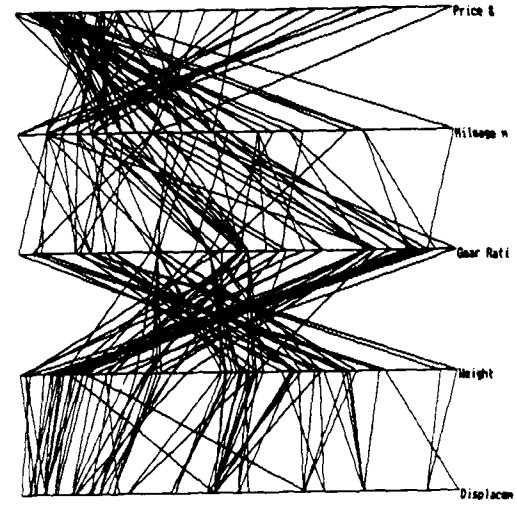


Figure 5.1 A parallel coordinate plot in five dimensions of automobile data. Note the negative correlation between gear ratios and weight.

engine providing considerable torque thus requiring a lower gear ratio. Conversely, a light car would tend to have a small engine providing small amounts of torque thus requiring a higher gear ratio.

Consider as well the relationship between weight and cubic inch displacement. In this diagram we have a considerable amount of approximate parallelism (relatively few crossings) suggesting positive correlation. This is a graphic representation of the fact that big cars tend to have big engines, a fact most are prepared to believe. Quite striking however is the negative slope going from low weight to moderate cubic inch displacement. This is clearly an outlier which is unusual in neither variable but in their joint relationship.

The relationship between miles per gallon and price is also perhaps worthy of comment. The left-hand side shows an approximate hyperbolic boundary while the right-hand side clearly illustrates the cross-over effect. This suggests for inexpensive cars or poor mileage cars there is relatively little correlation. However, costly cars almost always get relatively poor mileage while good gas mileage cars are almost always relatively inexpensive.

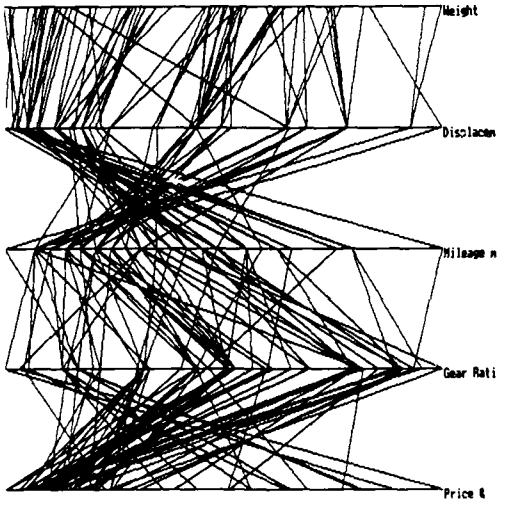


Figure 5.2 The second permutation of the five dimensional presentation of the automobile data. Notice the two classes of linear relations gear ratio and miles per gallon.

Turning to Figure 5.2, the relationship between gear ratio and miles per gallon is instructive. This diagram is suggestive of two classes. Notice that there are a number of observations represented by line segments tilted slightly to the right of vertical (high positive slope) and a somewhat larger number with a negative slope of about -1. Within each of these two classes we have approximate parallelism. This suggests that the relationship between gear ratios and miles per gallon is approximately linear, a believable conjecture since low gears = big engines = poor mileage while high gears = small engines = good mileage. What is intriguing, however, is that there seems to be really two distinct classes of automobiles each exhibiting a linear relationship, but with different linear relationships within each class.

Indeed in Figure 5.3, the third permutation, we are able to highlight this separation into two classes in a truly 5-dimensional sense. The shaded region in Figure 5.3 describes a class of vehicles with relatively poor gas mileage, relatively heavy, relatively inexpensive, relatively large engines and relatively low gear ratios. Figure 5.4 is a repeat of this graphic but with different shading highlighting a class of vehicles with

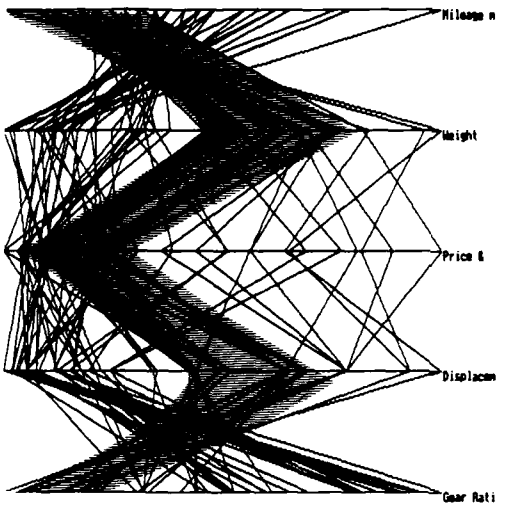


Figure 5.3 The third permutation of the five dimensional automobile data. Note the highlighting of the domestic automobile group.

relatively good gas mileage, relatively light weight, relatively inexpensive, relatively small engines and relatively high gear ratios. In 1980, these two characterizations describe respectively domestic automobiles and imported automobiles.

6. Graphical Extensions of Parallel Coordinate Plots. The basic parallel coordinate idea suggests some additional plotting devices. We call these respectively the Parallel Coordinate Density Plots, Relative Slope Plots and Color Histograms. These are extensions of the basic idea of parallel coordinates, but structured to exploit additional features or to convey certain information more easily.

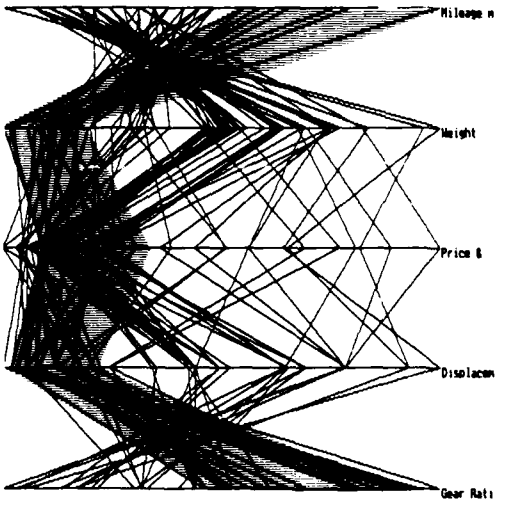


Figure 5.4 The third permutation showing highlighting of the imported automobile group.

6.1 Parallel Coodinate Density Plots. While the basic parallel coordinate plot is a useful device itself, like the conventional scatter diagram, it suffers from heavy overplotting with large data sets. In order to get around this problem, we use a parallel coordinate density plot which is computed as follows. Our algorithm is based on the Scott (1985) notion of

average shifted histogram (ASH) but adapted to the parallel coordinate context. As with an ordinary two dimensional histogram, we decide on appropriate rectangular bins. A potential difficulty arises because a line segment representing a point may appear in two or more bins in the same horizontal slice. Obviously if we have k n-dimensional observations, we would like to form a histogram based on k entries. However, since the line segment could appear in two or more bins in a horizontal slice, the count for any given horizontal slice is at least k and may be bigger. Moreover, every horizontal slice may not have the same count. To get around this, we convert line segments to points by intersecting each line segment with a horizontal line passing through the middle of the bin. This gives us an exact count of k for each horizontal slice. We construct an ASH for each horizontal slice (typically averaging 5 histograms to form our ASH). We have used contours to represent the two-dimensional density although gray scale shading could be used in a display with sufficient bit-plane memory. Because of our inability to reproduce color or gray-scale, we cannot give an example of a parallel coordinate density plot in this paper. Parallel coodinate density plots have the advantage of being graphical representations of data sets which are simultaneously high dimensional and very large.

6.2 Relative Slope Plots. We have already seen that parallel line segments in a parallel coordinate plot correspond to high positive correlation (linear relationship). As in our automobile example, it is possible for two or more sets of linear relationships to exist simultaneously. In an ordinary parallel coordinate plot, we see these as sets of parallel lines with distinct slopes. The work of Cleveland and McGill (1984b) suggests that comparison of slopes (angles) is a relatively inaccurate judgement task and that it is much easier to compare magnitudes on the same scale. The relative slope plot is motivated by this. In an n-dimensional relative slope plot there are $n-1$ parallel axes, each corresponding to a pair of axes, say x_i and x_j, with x_j regarded as the lower of the two coordinate axes. For each observation, the slope of the line segment between the pair of axes is plotted as a magnitude between -1 and $+1$. The maximum positive slope is coded as $+1$, the minimum negative slope as -1 and a slope of ∞ as 0. The magnitude is calculated as $\cos \eta$ where η is the angle between the x_j axis and the line segment corresponding to the observation. Each individual observation in the relative slope plot corresponds to a vertical section through the axis system. An example of a relative slope plot is given in Figure 6.1.

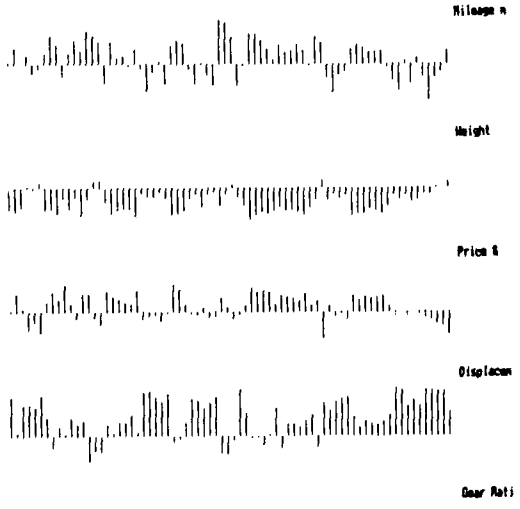


Figure 6.1 Relative slope plot of five dimensional automobile data. Data presented in the same order as in Figure 5.4

Notice that since slopes are coded as heights, simply laying a straightedge will allow us to discover sets of linear relationships within the pair of variables x_i and x_j.

6.3 Color Histograms. The basic set-up for the color histogram is similar to the relative slope plots. For an n-dimensional data set, there are n parallel axes. A vertical section through the diagram corresponds to an observation. The idea is to code the magnitude of an observation along a given axis by a color bin, the colors being chosen to form a color gradient. We typically choose 8 to 15 colors. The diagram is drawn by choosing an axis, say x_k, and sorting the observations in ascending order. Along this axis, we see blocks of color arranged according to the color gradient with the width of the block being proportional to the number of observations falling into the color bin. The observations on the other axes are arranged in the order corresponding to the x_k axis and color coded according to their magnitude. Of course, if the same color gradient shows up say on the x_m axis as on the x_k, then we know x_k is positively "correlated" with x_m. If the color gradient is reversed, we know the "correlation" is negative. We used the phrase "correlation" advisedly since in fact if the color gradient is the same but the color block sizes are different, the relationship is nonlinear. Of course if the x_m axis shows color speckle, there is no "correlation" and x_k is unrelated to x_m. Again we are unable to give an example of a color histogram in this paper because of our inability to reproduce color or gray-scale.

7. Implementations and Experiences. Our parallel coordinates data analysis software has been implemented in two forms, one a PASCAL program operating on the IBM RT under the AIX operating system. This code allows for up to four simultaneous windows and offers simultaneous display of parallel coordinates and scatter diagram displays. It offers highlighting, zooming and other similar features and also allows the possibility of nonlinear rescaling of each axis. It incorporates axes permutations and also includes Parallel Coordinate Density Plots, Relative Slope Plots and Color Histograms.

Our second implementation is under development in PASCAL for MS-DOS machines and includes similar features. In addition, it has a mouse-driven painting capability and can do real-time rotation of 3-dimensional scatterplots. Both programs use EGA graphics standards, with the second also using VGA or Hercules monochrome standards.

We regard the parallel coordinate representation as a device complementary to scatterplots. A major advantage of the parallel coordinate representation over the scatterplot matrix is the linkage provided by connecting points on the axes. This linkage is difficult to duplicate in the scatterplot matrix. Because of the projective line-point duality, the structures seen in a scatterplot can also be seen in a parallel coordinate plot. Moreover, the work of Cleveland and McGill (1984b) suggests that it is easier and more accurate to compare observations on a common scale. The parallel coordinate plot and the derivatives of it de facto have a common scale and so for example a sense of variability and central tendency among the variables are easier to grasp visually in parallel coordinates when compared with the scatterplot matrix. On the other hand, one might interpret all the ink generated by the lines as a significant disadvantage of the parallel coordinate plot. Our experience on this is mixed. Certainly for large data sets on hard copy this is a problem. When viewed on an interactive graphics screen particularly a high resolution screen, we have often found that individual points in a scatterplot can get lost because they are simply not bright enough. That does not happen in a parallel coordinate plot. However, if many points are plotted in monochrome, it is hard to distinguish between points. We have gotten around this problem by plotting

distinct points in different colors. In an EGA implementation, this means 16 colors. This is surprisingly effective in separating points. In one experiment, we plotted 5000 5-dimensional random vectors using 16 colors, and inspite of total overplotting, we were still able to see some structure. In data sets of somewhat smaller scale, we have implement a scintillation technique. With this technique, when there is overplotting we cause the screen view to scintillate between the colors representing the overplotted points. The speed of scintillation is is proportional to the number of points overplotted and by carefully tracing colors, one can follow an individual point through the entire diagram.

We have found painting to be an extraordinarily effective technique in parallel coordinates. We have a painting scheme that not only paints all lines within a given rectangular area, but also all line lying between to slope constraints. This is very effective in separating clusters. We also use invisible paint to eliminate observation points from the data set temporarily. This is a natural way of doing a subset selection.

Acknowledgements.

This research was funded by the Army Research Office under contract DAAL03-87-K-008, by the Air Force Office of Scientific Research under grant AFOSR-87–179 and by the National Science Foundation under grant DMS-8701931.

References

Asimov, Daniel (1985), "The grand tour: a tool for viewing multidimensional data," *SIAM J. Scient. Statist. Comput.*, 6, 128-143.

Carr, D. B., Nicholson, W. L., Littlefield, R., Hall, D. L. (1986), "Interactive color display methods for multivariate data," in *Statistical Image Processing and Graphics*, (Wegman, E. and DePriest, D., eds.), New York: Marcel Dekker, Inc.

Chernoff, H. (1973), "Using faces to represent points in k-dimensional space," *J. Am. Statist. Assoc.*, 68, 361-368.

Cleveland, W. S. and McGill, R. (1984a), "The many faces of the scatterplot," *J. Am. Statist. Assoc.*, 79, 807-822.

Cleveland, W. S. and McGill, R. (1984b), "Graphical perception: theory, experimentation, and application to the development of graphical methods," *J. Am. Statist. Assoc.*, 79, 531-554.

Diaconis, P. and Friedman, J. (1983), "M and N plots," in *Recent Advances in Statistics*, 425-447, New York: Academic Press, Inc.

Dimsdale, B. (1984), "Conic transformations and projectivities," IBM Los Angeles Scientific Center Report #6320-2753.

Fienberg, S. (1979), "Graphical methods in statistics," *Am. Statistician*, 33, 165-178.

Friedman, J. and Tukey, J. W. (1973), "PRIM-9" a film produced by Stanford Linear Accelerator Center, Stanford, CA Bin 88 Productions, April, 1973.

Friedman, J. and Tukey, J. W. (1974), "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, C-23, 881-889.

Hartigan, John A. (1975), *Clustering Algorithms*, New York: John Wiley and Sons, Inc.

Griffen, H. D. (1958), "Graphic computation of tau as a coefficient of disarray," *J. Am. Statist. Assoc.*, 53, 441-447.

Holmes, S. D. (1928), "Appendix B: a graphical method for estimating R for small groups," 391-394 in *Educational Psychology* (Peter Sandiford, auth.), New York: Longmans, Green and Co.

Inselberg, A. (1985), "The plane with parallel coordinates," *The Visual Computer*, 1, 69-91.

Scott, D. W. (1985), "Average shifted histograms: effective nonparametric density estimators in several dimensions," *Ann. Statist.*, 13, 1024-1040.

GRAPHICAL REPRESENTATIONS OF MAIN EFFECTS AND INTERACTION EFFECTS IN A POLYNOMIAL REGRESSION ON SEVERAL PREDICTORS

William DuMouchel, BBN Software Products Corporation

Abstract

The table of coefficients from a polynomial regression analysis having several predictors is hard to interpret because its focus is on the terms in the fitted equation, rather than on the variables used to define those terms. Methods for graphically comparing the effects of each predictor to each other and to the residuals are introduced and discussed. The techniques are easy to implement and to interpret, and have been generalized to provide graphical summaries of interaction effects.

1. Introduction

Partial residual plots (also known as component-plus-residual plots) are useful diagnostic tools in multiple regression analysis. Mallows (1986) discusses them and suggests an extension of the technique, which he calls an augmented partial residual plot, designed to reveal a nonlinear effect in a regression model. This paper introduces generalizations of such plots which are designed to help a data analyst interpret the fit to an arbitrary response surface model (RSM), a regression equation in the form of a polynomial in several variables. This new technique, called an *adjusted-Y plot*, can also be used to help diagnose nonlinearity of a regression function with respect to one of the predictors, and in fact, if the regression model being fitted is additive and linear in the predictors, the adjusted-Y plot reduces to the partial residual plot. However, the adjusted-Y plot is useful for an arbitrary polynomial RSM, and the emphasis of this technique is not so much to diagnose nonlinearity as to visualize the nonlinearity which has already been incorporated into the RSM, with a secondary goal of diagnosing deviations from the assumptions of the RSM. The adjusted-Y plot is especially useful as the foundation for other graphical techniques for comparing the effects of the different predictor variables in the RSM, and for helping the data analyst visualize the size and significance of interaction effects.

Partial residuals. Suppose that a linear regression model with J predictors, of the form

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \ldots + b_J x_{iJ} + e_i \; ; \; i = 1, \ldots, n,$$

has been fit by least squares, where the b's are the estimated coefficients and the e's are the residuals. Suppose it is desired to focus on one of the predictors, say $x_1 = (x_{i1} \; ; \; i = 1, \ldots, n)$, and check the assumptions of constant variance and linearity with respect to that predictor. The partial residuals (pr) with respect to x_1 are defined as

$$pr_{i1} = \bar{y} + b_1 (x_{i1} - \bar{x}_1) + e_i , \qquad i = 1, \ldots, n,$$

where $\bar{y}$ and $\bar{x}_1$ are means, and b_1 and the e_i are taken from the full regression. The plot of pr_{i1} vs x_{i1} has the advantage that it displays both the signal coming from x_1 (the term $b_1(x_{i1} - \bar{x}_1)$) and the noise (the term e_i) as they occur in the regression on all J predictors. This plot is to be distinguished from the "added variable" or "partial regression" plot, which is similar to the partial residual plot except that the values plotted on the horizontal axis are the residuals $(x_{i1} - \hat{x}_{i1})$, based on a regression of x_1 on the remaining J-1 predictors, rather than the x_{i1} themselves.

Augmented partial residuals. Mallows (1986) suggested that nonlinearity in the relationship between y and x_1 can be better detected by adding $(x_{i1} - \bar{x}_1)^2$ to the regression equation and then replacing pr_{i1} by

$$apr_{i1} = \bar{y} + b_1(x_{i1} - \bar{x}_1) + c[(x_{i1} - \bar{x}_1)^2 - ave] + e_i ,$$

where b_1 and c are coefficients and the e's are residuals from the augmented regression model, and where ave is the average of $(x_{i1} - \bar{x}_1)^2$ in the sample. The augmented partial residual plot is most effective, compared to the simple partial residual plot, when one or more of the other predictors are correlated with the term $(x_{i1} - \bar{x}_1)^2$.

Adjusted-Y plots. Suppose that a response surface model equation is represented as

$$y_i = F(x_{i1}, x_{i2}, \ldots, x_{iJ}) + e_i ,$$

where F is the fitted polynomial and the e's are the residuals from the regression. For any one of the predictors, say x_1, define an *adjusted-fit* function over the range of x_1 as

$$f_1(x) = \frac{1}{n} \Sigma_k \, F(x, x_{k2}, \ldots, x_{kJ}), \qquad (1)$$

and define an *adjusted-Y* variable for the i^{th} observation as

$$y_{i1}^{adj} = f_1(x_{i1}) + e_i . \qquad (2)$$

As proved in Section 5.2, if F is of the form $b_0 + b_1 x_1 + F^*(x_2, \ldots, x_J)$, then every $y_{i1}^{adj} = pr_{i1}$. Also, if F is of the form $b_0 + b_1 x_1 + b_{11} x_1^2 + F^*(x_2, \ldots, x_J)$, then every $y_{i1}^{adj} = apr_{i1}$. The adjusted-Y plot is a generalization of the partial residual and the augmented partial residual plots which is useful for response surface models having arbitrary power and interaction terms.

2. Example Usage of the Adjusted-Y Plot

2.1 Data and Standard Analysis

The data used in this example were taken from Andrews and Herzberg (1985, p.355-6) and consist of measurements on 42 apple trees in an agricultural experiment. The response is the mean weight (Wt) of mature apples on each tree, which was considered to be a proxy for the relative freedom of the apples from a disease which shrivels them. Several variables were measured and reported by Andrews and Herzberg (1985) and by the original researchers Ratkowsky and Martin (1974), but this example will use just three of them, the concentrations, in parts per million, of three minerals in the apples of each tree. The three concentrations, labeled K, Pn, and Ca respectively, were used to form a response surface model to describe the association between the mineral concentrations and the mean weight variable. After some preliminary modeling, an equation of the form

$$Wt = b_0 + b_1 Pn + b_2 K + b_3 Ca + b_4 Pn*K + b_5 Pn*Ca + b_6 K^2 + e \quad (3)$$

was considered adequate for describing the data. Figure 1 shows the table of coefficients and some related statistics for this regression model. Figure 2 shows a scatterplot of the absolute values of the studentized residuals versus the fitted values from this regression, with a lowess fit to the points showing no pattern indicating a violation of the usual assumptions of regression models. Figure 3 shows a scatterplot of Wt versus K, with three different symbols used to denote points falling within three ranges (low, medium, and high) of the variable Pn.

Least Squares Coefficients, Response WT, Model PN_K_CA

0 Term	1 Coeff.	2 Std. Error	3 T-value	4 Signif.
1 1	[illegible]	129.179267	1.90	0.0659
2 PN	-0.171712	0.076722	-2.24	0.0317
3 K	0.009638	0.014979	0.64	0.5241
4 CA	-0.377748	0.244072	-1.55	0.1307
5 PN*K	0.000012	0.000006	1.94	0.0607
6 PN*CA	0.000117	0.000123	0.95	0.3492
7 K**2	-0.000001	9.523644e-07	-1.20	0.2386

No. cases = 42 R-sq. = 0.7532 RMS Error = 9.462
Resid. df = 35 R-sq-adj. = 0.7109 Cond. No. = 441.6

Figure 1. Table of coefficients and related output for the example data.

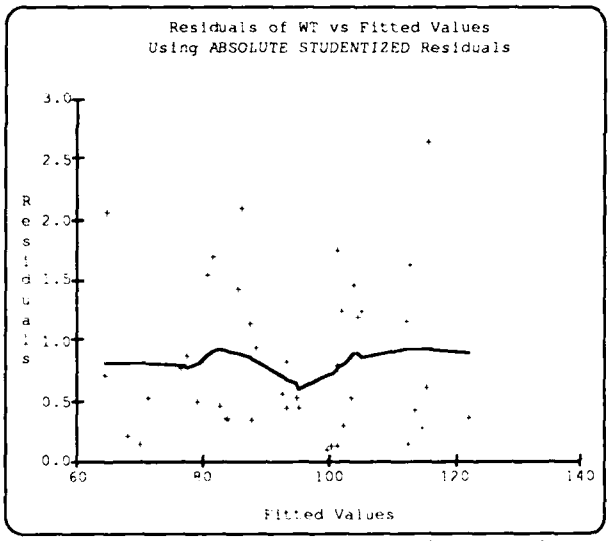


Figure 2. Absolute residual plot with lowess curve for the example data.

Comparing Figure 1 with Figure 3, the difficulty of interpreting the table of coefficients from a response surface model becomes evident. Although the scatterplot of the raw data in Figure 3 seems to show a definite relationship between Wt and K, none of the three terms in the table of coefficients that contain K as a factor has a significant coefficient, although the term Pn*K is borderline. In fact, a casual glance at the table of coefficients is not enough to confirm that the fitted value of Wt increases with K, since complicated comparisons of the relative contributions of the linear, quadratic and interaction terms are required. If all three mineral concentrations had been standardized to have mean 0 and variance 1, the task of sorting out the effects of each mineral from the table of coefficients would be somewhat eased, but correlations among the minerals can prevent an easy interpretation no matter how they are scaled.

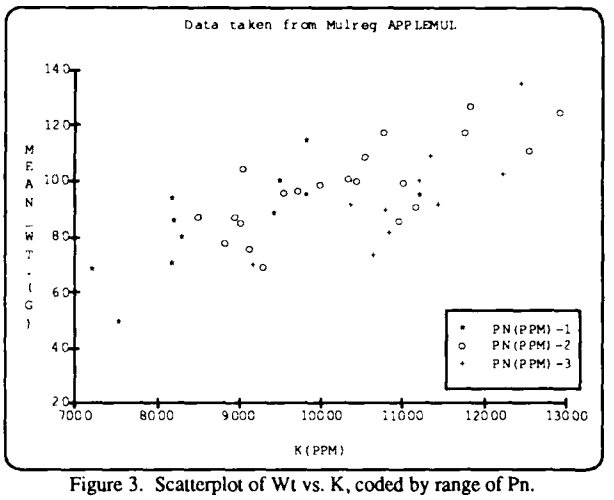


Figure 3. Scatterplot of Wt vs. K, coded by range of Pn.

A closer look at Figure 3 shows that the predictors K and Pn are indeed correlated, since the pattern of symbols denoting approximate values of Pn on the plot shows that low and high values of K tend to be associated with low and high values of Pn, respectively. So there is ambiguity in Figure 3; the apparent trend of Wt with K could be due partially to confounding with the effect of Pn, and the apparent linearity of the trend could also be an artifact of the confounding. The scatter of the points in Figure 3 about this trend is also ambiguous, since it is due partly to the error term from the regression and partly to the effects of the other two predictors.

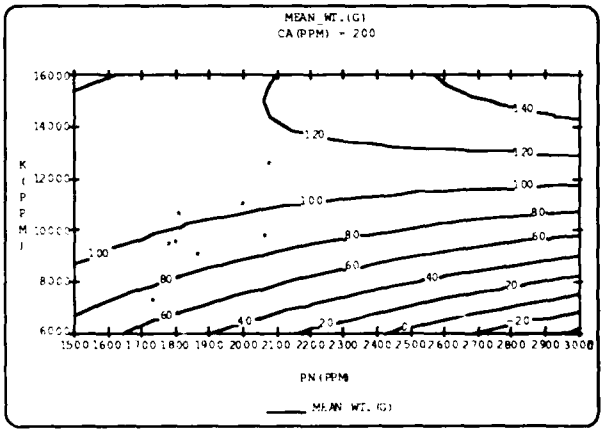


Figure 4. Contour plot of part of the fitted response surface. Plotted points are locations of raw data.

Figure 4 shows a contour plot of the fitted surface versus K and Pn at the point Ca=200. Contour plots are frequently used to study fitted response surfaces, but they have some limitations. Many people without a technical background find contour plots more difficult to interpret than the basic X-Y plot. There is no measure of uncertainty on the standard contour plot: no residuals to show where there might be lack of fit to the model, and no error bars to show the magnitude of the sampling error inherent in the contours. Each contour plot must fix all but two of the predictors, so only a small slice of the design space is portrayed on contour plots of models having several predictor variables.

2.2 Adjusted-Y Plots

Figures 5 and 6 show the adjusted-Y plots for the variables Pn and K, respectively. Figure 5 shows the average (over the n=42 sample points) of fitted Wt versus Pn, namely the straight line $f_1(Pn)$, with the residuals from the response surface fit added to form the ordinates, the values of adjusted-Wt defined by (2). Figure 6 shows the average of fitted Wt versus K, namely the parabola $f_2(K)$, with the same residuals added to form the second set of adjusted-Wts. It is instructive to compare Figure 6 with Figure 3. The ambiguities of Figure 3 have been cleared up in Figure 6. The dependence of Wt on K is portrayed in Figure 6 clear of any confusion with the effects of Pn or Ca. The adjusted-Y plot communicates the magnitude of the curvature and the strength and direction of the overall trend more powerfully than does the table of coefficients. Figure 6 also makes possible a comparison of the relative magnitudes of the variation due to K versus the unexplained variation in Wt. Figure 3 is misleading in this comparison.

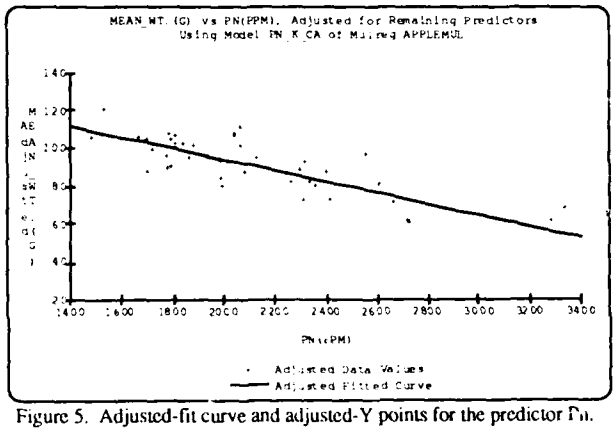


Figure 5. Adjusted-fit curve and adjusted-Y points for the predictor Pn.

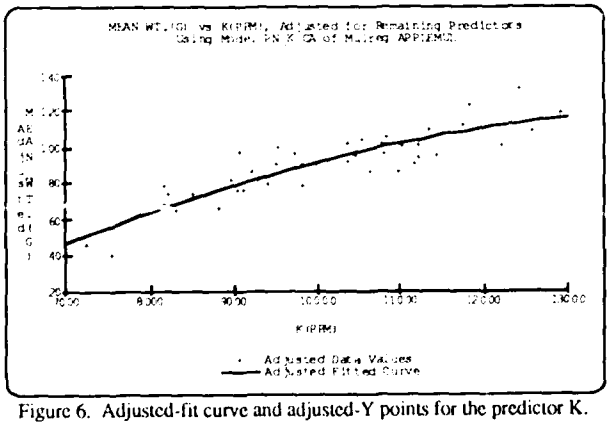


Figure 6. Adjusted-fit curve and adjusted-Y points for the predictor K.

Since the curve in Figure 6, $f_2(K)$, is the average of n=42 parabolas, and since the model does contain an interaction term between K and Pn, it is possible that for some values of Pn the behavior of the fitted function will be quite different from that of $f_2(K)$. But the average behavior, at least, is easily visualized, standardized to the distribution of the other two predictors in the sample. And any K-regions of lack of fit of the points to the model are easily identified. In Section 4 a method of displaying the interaction effects in response surface models is described.

3. Standard Errors for Adjusted-Fits and for Average Effects

3.1 Development of Standard Errors.

We now shift attention from the adjusted-Y values, which are interpreted much like partial residuals, to the adjusted-fit curves. If the fitted response surface F is linear and additive in every predictor, and contains a constant term, then the j^{th} adjusted-fit curve is just

$$f_j(x) = \bar{y} + b_j (x - \bar{x}_j),$$

where b_j is the coefficient of x_j in the multiple regression. In this case the variance of $f_j(x)$ would be estimated by

$$\frac{mse}{n} + V(b_j)\,(x - \bar{x}_j)^2 ,$$

where mse is the mean squared error of the residuals and $V(b_j)$ is the usual estimate of the variance of the regression coefficient, based on the inverse of the $\mathbf{X}^t\mathbf{X}$ matrix from the regression.

In the general case of a polynomial response surface in J variables, the j^{th} adjusted-fit curve is a polynomial of degree p_j

$$f_j(x) = \Sigma_{k=0}^{p_j} B_{jk}\, x^k ,$$

where p_j is the largest power of x_j which occurs in F. The coefficients B_{jk} are linear combinations of the b's, the coefficients of F. For example, using the RSM of (3),

$$f_2(K) = B_{20} + B_{21}\, K + B_{22}\, K^2,$$

$$B_{20} = b_0 + b_1 \overline{Pn} + b_3 \overline{Ca} + b_5 \overline{Pn*Ca} ,$$

$$B_{21} = b_2 + b_4 \overline{Pn} ,$$

$$B_{22} = b_6 .$$

where the constants $\overline{Pn}$, $\overline{Ca}$ and $\overline{Pn*Ca}$ are averages of the three corresponding terms over the n sample points. Thus, if **b** is the vector of least squares coefficients and if $\mathbf{B}_j$ is the vector $(B_{j0}, B_{j1}, \ldots)^t$, then there is a matrix $\mathbf{A}_j$, with elements formed from averages of predictor variable terms, which transforms **b** to $\mathbf{B}_j$:

$$\mathbf{B}_j = \mathbf{A}_j\, \mathbf{b} ,$$

$$\mathbf{C}_j = \text{estimated covariance matrix of } \mathbf{B}_j ,$$

$$= \mathbf{A}_j (\mathbf{X}^t\mathbf{X})^{-1}\, \mathbf{A}_j^t\, mse.$$

3.2 Confidence Intervals for Adjusted Effects of Variables

Once the covariance matrix of each $\mathbf{B}_j$ is available, it is easy to obtain standard errors and confidence intervals for the functions $f_j(x)$ at any point x. The curves in Figures 5 and 6 could have error bars or even upper and lower confidence curves drawn about them on the figures. Such confidence intervals may not often be useful, since the height of the adjusted-fit curve at any point is not a predicted response at any particular design point, but is instead an average of predictions at n design points. A more useful application of these covariances is for the computation of confidence

intervals for contrasts based on the difference of two values, $f_j(x) - f_j(x')$.

As an example, look at the adjusted-fit curve in Figure 6, $f_2(K)$. Within the range of the data the minimum value of f_2 is $f_2(7240)=51$, and the maximum value of f_2 is $f_2(12910)=116$. So 116 - 51 = 65 is the estimated increase in Wt when K changes from 7240 to 12910, adjusted for all the other predictors. A confidence interval about this difference is derived as follows:

Define $\mathbf{x}$ as the row vector $(1, x, x^2, \ldots)$, and define $\mathbf{x}'$ analogously. Then

$$f_j(x) - f_j(x') = (\mathbf{x} - \mathbf{x}')\,\mathbf{B}_j\,,$$

$$v_j = \text{estimated variance of } [f_j(x) - f_j(x')]\,,$$

$$= (\mathbf{x} - \mathbf{x}')\,\mathbf{C}_j\,(\mathbf{x} - \mathbf{x}')^t\,.$$

Therefore a confidence interval for the increase in mean response associated with changing the j^{th} variable from x to x' is

$$f_j(x) - f_j(x') \pm t(df,\ 1\text{-}\alpha/2)\,\sqrt{v_j}\,, \qquad (4)$$

where $t(df, 1\text{-}\alpha/2)$ is a tabled student's-t percentile with degrees of freedom equal to the degrees of freedom of mse.

Figure 7 graphs these confidence intervals for the effects of the three predictor variables in the RSM for the apple data. In each case, the values of x and x' used are the maximum and minimum values, respectively, of the predictor in the sample. (Section 5.1 provides the rules for choosing x and x' in general, and also discusses the choice of tabled percentile for the width of the interval.)

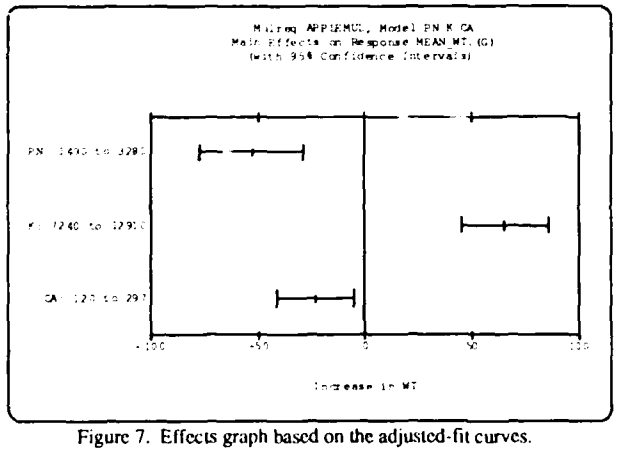


Figure 7. Effects graph based on the adjusted-fit curves.

Compare Figure 7 with Figure 1, the table of coefficients, as summaries of the RSM analysis. The information in Figure 7 is tremendously more accessible. You can see at a glance that Pn has a negative effect of about 55 grams, K has a positive effect of about 65 grams, Ca has a negative effect of about 25 grams, and that all three effects are statistically significant. (The word "effect" as used here is not intended to imply a causal effect, merely an associated change in mean response.) The table of coefficients is quite opaque by comparison. It is practically impossible to tell which variables have effects in which directions without elaborate calculation, much less gauge the relative significance of the three predictors. The problem with the table of coefficients as a summary of the analysis is that it focuses on the terms of the model, not on the variables of the model.

Figure 4, a contour plot of the fitted RSM, although it does focus on the variables, is still much less effective than Figure 7 as a summary of the analysis. Figure 4 gives no information about the effect of Ca, and no information about the statistical significance of any of the effects. And it is just plain harder to read.

This is not to say that you should never look at tables of coefficients or contour plots of RSM fits, just that the graph of effects as here defined is a valuable addition to the statistician's toolbox, especially in conjunction with the adjusted-Y plots discussed previously and the interaction graphs discussed in the next section.

4. Interaction Graphs

4.1 The Bivariate Adjusted-Fit Function

In order to explore the interaction inherent in a fitted RSM equation, we extend the definition of the adjusted-fit function of (1) to the bivariate case. Suppose we are interested in the fitted relationship as a function of two of the predictors, say x_1 and x_2, after adjusting for all other predictors. As before, let $F(x_1, x_2, x_3, \ldots, x_J)$ be the fitted RSM equation, and define

$$f_{12}(x, z) = \frac{1}{n}\Sigma_k\, F(x, z, x_{k3}, \ldots, x_{kJ}).$$

In the case of the example model (3),

$$f_{12}(Pn, K) = (b_0 + b_3\,\overline{Ca}) + (b_1 + b_5\,\overline{Ca})Pn + b_2\,K + b_4\,Pn{*}K + b_6\,K^2\,.$$

As in the case of the univariate adjusted-fit function, the coefficients of f_{12} are simple functions of the coefficients of F and certain moments of the predictors which are being averaged out. It is similarly straightforward to compute the variance of $f_{12}(x, z)$ at any value of (x, z), or the variance of any difference of the form $[f_{12}(x, z) - f_{12}(x', z')]$ for any pairs of values.

4.2 Displaying Interaction Effects

Figure 8 shows how the effect of an interaction term in a RSM can be displayed in a graph analogous to the effects graph of Figure 7. The top bar in Figure 8 repeats the top bar in Figure 7, a confidence interval for the effect of Pn, namely $f_1(3280) - f_1(1490)$. The next three bars in Figure 8 display confidence intervals for $f_{12}(3280, K) - f_{12}(1490, K)$, for three values of K. That is, the same contrast in Pn is repeated assuming K is fixed, for various values of K. By comparing these three intervals, you can see the direction, magnitude, and significance of the interaction between Pn and K in their effect on Wt, as measured by the RSM. Since the midpoints of the intervals move to the right as K increases, the interaction is positive. The magnitude of the interaction is about the same as the main effect of Pn, since at the largest value of K the effect of Pn is almost exactly 0, while at the minimum value of K the effect of Pn is about double its average value. And the interaction is on the borderline of being statistically significant, since the confidence intervals for the effect of Pn at the high and low values of K barely overlap. (In this case the judgement of statistical significance is merely approximate,

since the overlapping of the two confidence intervals does not rule out finding a significant difference. But an approximate indication of the sampling error is clearly communicated by the graph.)

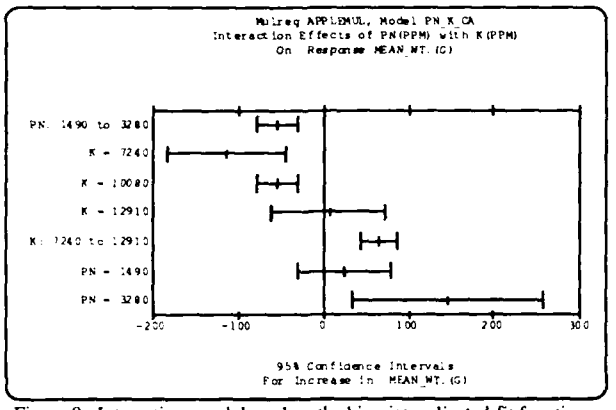


Figure 8. Interaction graph based on the bivariate adjusted-fit function.

The bottom three confidence intervals in Figure 8 provide the dual interpretation of the interaction between Pn and K. First the main effect of K, measured as $f_2(12910) - f_2(7240)$, is graphed exactly as in the middle interval of Figure 7. Below it are confidence intervals for $[f_{12}(Pn, 12910) - f_{12}(Pn, 7240)]$, for the extreme values of Pn in the sample. Comparison of these intervals leads to the same interpretation as before, but with emphasis on how the effect of K changes as a function of Pn, rather than vice-versa.

As a device for visualizing interaction, Figure 8 has advantages over Figure 4, the contour plot. In order to figure out the direction of the interaction from the contour plot, you can notice that the contours are more closely spaced in the vertical (K) direction where Pn is large than where Pn is small. This indicates that K has a greater effect when Pn is large than when Pn is small. But perceiving the magnitude of the interaction from Figure 4 is even more difficult, while there is no indication at all of statistical significance.

The table of coefficients in Figure 1, on the other hand, does display the direction and significance of the Pn*K term (p=.06), but the magnitude of the interaction effect compared to the other effects is hard to see from the table alone. And if the model were expanded to contain cubic terms like $Pn*K^2$, then even the significance of the interaction could become obscured in the table by the correlations between the various terms of the model.

One frequently recommended method for visualizing the interaction between two factors on a response is to graph the response versus one of the factors separately for different levels of the other factor. The scatterplot in Figure 3 is such a plot, but the plot in Figure 9 better illustrates the idea by overlaying smoothed lowess curves over each of the three sets of points. The curve based on the largest values of Pn is steepest, confirming the interaction effect we have been studying. This method of displaying interaction is particularly effective if the data come from a two-level orthogonal experimental design, since the plot then consists of just a pair of straight lines, and, if the design has resolution at least 5, the interaction is not confounded with effects of other variables. The interaction graph of Figure 8 complements that of Figure 9. Figure 8 is model-based while Figure 9 is an exploratory graph based on the raw data. Figure 8 contains more precise information on the extent and significance of the interaction, while Figure 9 displays response values directly, rather than being based on differences of responses.

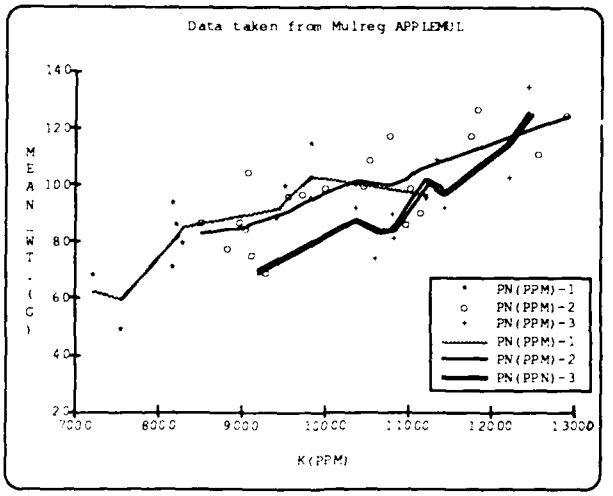


Figure 9. Scatterplot as in Figure 3, with lowess curves added.

5. Discussion

This section discusses several issues related to the implementation of these methods, and concludes with a proof that the adjusted-Y plot is equivalent to the partial residual plot when the model is additive with respect to the selected predictor.

5.1 Implementation Issues

In order to create the special plots introduced here, a multiple regression program must have data structures that enable the system to analyze each term of the RSM, and to determine which variables are involved. The Mulreg program, from BBN Software Products Corporation, lets the user specify, fit, and compare different models; the definition of a model includes a list of terms containing such information. Using this information, it is relatively simple to sort the terms and compute the adjusted-fit functions by calculating the B's from the b's and certain sample moments, as discussed in Section 3.1. The confidence intervals for the effects and interaction graphs can then be computed as described in Section 3.2. The following paragraphs discuss the rationale for several of the choices which the system makes in forming these confidence intervals.

Choice of comparison values in the effects graph. The effects graph of Figure 7 is based on three adjusted-fit curves, one for each predictor. The center of the j^{th} confidence interval is of the form $f_j(x) - f_j(x')$, where x and x' are chosen separately for each j so that:

(1) x and x' are within the sample range of $\mathbf{x}_j$, and
(2) the absolute difference $|f_j(x) - f_j(x')|$ is maximized, and
(3) if $\mathbf{x}_j$ is measured on a numerical scale, x'<x.

If $f_j(x)$ is linear, these constraints imply that $x' = \min(x_{1j}, \ldots, x_{nj})$, $x = \max(x_{1j}, \ldots, x_{nj})$. If $f_j(x)$ is quadratic, the system determines the extreme point of f_j as $x'' = -B_{j1} / 2B_{j2}$.

If $x'' < \min(x_{1j}, \ldots, x_{nj})$ or if $x'' > \max(x_{1j}, \ldots, x_{nj})$, then x and x' are chosen as for the linear case. Otherwise x'' replaces either $\min(x_{1j}, \ldots, x_{nj})$ or $\max(x_{1j}, \ldots, x_{nj})$, so that condition (2) above is satisfied. If $f_j(x)$ is a cubic or higher degree polynomial, then the system evaluates $f_j(x)$ at $\min(x_{1j}, \ldots, x_{nj})$ and $\max(x_{1j}, \ldots, x_{nj})$ and at nine equally spaced points in between, and then chooses x and x' to maximize $|f_j(x) - f_j(x')|$ from among these eleven points, resulting sometimes in an approximate maximization of $|f_j(x) - f_j(x')|$.

If x_j is a categorically scaled variable, x and x' are the two categories having the extreme values of f_j.

The interaction graph described in Section 4.2 and shown in Figure 8 repeats the comparisons of the effects graph for two variables which share one or more interaction terms in the model. The choice of points for the second variable at which the contrasts for the first variable are repeated is made as follows: If the second variable is categorically scaled, the contrast is repeated at every level of the variable. If the second variable is continuous but enters the model only linearly, the contrast is repeated only at its minimum and maximum values. If the model contains higher powers of the variable, the contrast is repeated at the minimum, maximum, and midrange of its values in the sample.

Simultaneous confidence intervals. The confidence intervals within a single effects graph or interaction graph are not joint confidence intervals. The stated degree of confidence pertains to each interval separately. However, if a particular adjusted-fit function has more than one degree of freedom for contrasting x-values, as will happen if the function of a continuous variable is quadratic or higher order, or if a categorically scaled predictor has three or more categories, then the Mulreg program adjusts the confidence interval to account for the *post-hoc* manner in which x and x' are selected.

If $f_j(x)$ is a polynomial of degree p, then the Scheffé technique of replacing the $t(df, 1-\alpha/2)$ percentile in equation (4) by the percentile $\sqrt{p\, F(p, df, 1-\alpha)}$ is used. In the case of a categorically scaled predictor, a Bonferonni adjustment is made: the $t(df, 1-\alpha/2)$ percentile is replaced by the percentile $t(df, 1-\alpha/m(m-1))$, where m is the number of categories being compared.

Confidence intervals in the interaction graph described in Section 4.2 are computed using the same tabled critical values as the effects graph of the same contrast. In Figures 7 and 8, the intervals displaying contrasts with respect to K use the Scheffé method with 2 degrees of freedom, while the other intervals use the percentile $t(df, 1-\alpha/2)$, since the fitted function is linear in Pn and Ca.

Choice of error term. Mulreg usually uses the mean squared error of the residuals (mse) in the computation of the confidence intervals for the effects graph and the interactions graph, as discussed in Section 3. There are three circumstances in which another quantity is substituted for the residual mean squared error in the formulas.

First, if the data for the multiple regression contains replications, the system computes a mean square for "pure error" based on the response variation within groups of points having the same set of x-values. The confidence intervals in the Mulreg effects graphs and interactions graphs use the pure error mean square instead of the residual mean square whenever there are at least four degrees of freedom for pure error and the usual F-test for lack of fit is significant at the 10% level.

Second, if a model contains an interaction between a fixed effect and a random effect, then the confidence interval for the contrast of levels of the fixed effect will use the interaction mean square rather than the mean square residual.

Third, if a robust bisquare regression is being used rather than a least squares fitting algorithm, the robustly estimated coefficients and a robust version of the mean square error are substituted into the formulas.

Transformations. If the response variable has been transformed, the adjusted-Y, the effects graph, and the interaction graph are all computed and displayed on the transformed metric. In order to make these graphs more interpretable in such cases, the program can use a "matched" scaling of the response transformation, as recommended by Hoaglin et. al. (1983, section 4E).

5.2 *Proof of Equivalence to Partial Residuals*

Suppose that x_1 enters the model additively, so that the fitted least squares model is of the form

$$y_i = F_1(x_{i1}) + F^*(x_{i2}, \ldots, x_{ij}) + e_i, \qquad i=1, \ldots, n.$$

Then, using (1) and (2), the adjusted-Y values with respect to the first predictor variable are

$$y_i^{adj} = F_1(x_{i1}) + \bar{F}^* + e_i,$$

while the corresponding partial residuals (which are augmented partial residuals if F_1 is not linear) are

$$pr_i = \bar{y} + (F_1(x_{i1}) - \bar{F}_1) + e_i.$$

Comparing these formulas, we note that they are equal if

$$\bar{y} = \bar{F}_1 + \bar{F}^*,$$

which is the requirement that the average of the fitted values from the regression equals the average value of the response. As is well known, this will be true whenever a least squares regression model contains a constant term, or whenever some linear combination of the predictor terms is constant for all n cases. If the model cannot be reparametrized to contain a constant term, then, depending on how the partial residuals are defined, they may differ by a constant amount from the adjusted-Y values.

6. References

Andrews, D. F., and Herzberg, A. M. (1985) *Data*. New York: Springer Verlag.

Hoaglin, D., Mosteller, F. and Tukey, J., eds. (1983) *Understanding Robust and Exploratory Data Analysis*. New York, Wiley Interscience.

Mallows, C. L. (1986) Augmented Partial Residuals. *Technometrics* 28:313-320.

Ratkowsky, D.A. and Martin, D. (1974) The use of multivariate analysis in identifying relationships among disorder and mineral element content in apples. *Aust. J. Agric. Res.* 25:783-790.

V. COMPUTATIONAL ASPECTS OF SIMULATED ANNEALING

COMPUTATIONAL EXPERIENCE WITH GENERALIZED SIMULATED ANNEALING

Daniel G. Brooks and William A. Verdini, Arizona State University

ABSTRACT

Stochastic optimization procedures have been shown to be efficient methods for finding global extrema of objective functions. In this article we report computational results obtained using the generalized simulated annealing method on a set of standard global optimization test problems. The results are compared to those obtained using a self-regulating mechanism which chooses a random step distribution based on the local topography and the currently specified annealing temperature.

INTRODUCTION

The problem of finding the global extremum (assumed to be a minimum here) of a real-valued function has been an important one for a long time. There has been a recent increase in interest in solving global optimization problems using stochastic methods which, though computationally intensive, are efficient because of the increased speed of computation now available. These methods combine some form of sampling (usually random) and local search procedures. The better-known stochastic optimization methods have some very attractive behavioral properties and have proved to be efficient search procedures over a wide range of objective function topographies, including problems with high dimensionality and multiple extrema.

A stochastic method based on the simulation of the cooling of a liquid substance was shown to be useful for function optimization by Kirkpatrick, Gelatt, and Vecchi (KGV)(1983). Called "simulated annealing," the method has proven to be very useful for solving large combinatorial problems as documented by KGV and others, including NP-hard problems like Bonomi and Lutton's (1984) work with the traveling salesman problem. The method also has attractive theoretical properties for discrete spaces; Lundy and Mees (1986), Hajek (1986), and Geman and Geman(1984) all prove convergence of the algorithm under various assumptions on classes of NP-hard problems. The extensive bibliography compiled by Golden (to appear in 1988) contains numerous references of applications to combinatorial problems.

Simulated annealing applied to functions of continuous variables behaves much like a random walk with a bias, but lacks reasonable convergence behavior in many applications. Certain modifications can be made to hasten the method's convergence, such as the stepwise parameter adjustment of Vanderbilt and Louie (1984), who report solution times for their method. The study reported here investigates the behavior of the generalized simulated annealing (GSA) method introduced by Bohachevsky, Johnson, and Stein (1986), which uses the current value of the function to control the random process; various aspects of the method's behavior over a range of test problems with continuous variables are shown. Section 2 presents the simulated annealing algorithm and its generalization, and Vanderbilt and Louie's self-regulating simulated annealing (SRSA) algorithm. Section 3 gives the results of its application to the test problems, and Section 4 gives a summary with comments.

APPROACHES TO SIMULATED ANNEALING

"Annealing" refers to the process in which a substance is first melted, then the temperature is lowered slowly. The substance is allowed to spend a lot of time at temperatures near the freezing point of the substance, thereby allowing the atoms in the substance to arrange themselves into configurations with the lowest potential energy. The desire is to achieve a "ground state" (lowest potential energy) arrangement of atoms at each temperature. This ground state configuration occurs when the potential energy (function) is at its global minimum for all possible arrangements of atoms at that temperature. KGV give an interesting history of how Metropolis et al. (1953) developed an algorithm to simulate this annealing process for any particular substance. Starting with a substance with an arrangement of atoms at potential energy E, the Metropolis algorithm simulates

- a new arrangement of atoms resulting in a change in energy, denoted ΔE;
- If ΔE is negative, accept the arrangement by letting that be the new arrangement of atoms for the substance;
- If ΔE is positive, accept the arrangement with probability of $\exp(-\Delta E/K_B T)$

  where T is the temperature of the substance and K_B is the Boltzmann constant.

The simulation "moves" from configuration to configuration, following a

random walk with a bias to lower energy values, since the probability of acceptance of lower-energy arrangements is greater. The simulation assumes the system evolves into a Boltzmann distribution.

The analogy to more general applications is clear: the energy function is any objective function, the arrangement of atoms is the combination of independent variable values, and the rearrangement of atoms is equivalent to the iterative improvement of function values by changing variable values. The usefulness of simulated annealing as a function optimization procedure is that it can move to detrimental function values in its optimization search, which prevents it from being trapped in local minima. In addition, the implementation of the search for the global minimum does not require any derivatives, only function evaluations, making it both analytically and computationally convenient.

This standard annealing method is handicapped in function optimization, however, because there is no "cooling" (referred to as an annealing schedule); that is, the temperature of the substance remains fixed and, therefore, excessive numbers of moves are made in searching for minimum-energy configurations. The generalized simulated annealing method provides a gradual (though not necessarily monotonic) decline of temperature values thereby reducing the probability of acceptance of a higher-energy, and detrimental, point as the function values approach the (estimated or known) global minimum of the function. This is achieved by automatically setting the acceptance probability according to the function topography. The change in position is governed by a specified acceptance probability which depends on the parameters of an acceptance probability function. This function decreases the probability of moving to a new location as the algorithm progresses.

Simulated annealing has an exponential acceptance probability function so that the probability of moving from the location at the i-th function evaluation to the new location corresponding to the (i+1)-th evaluation is

$$p_i = \exp(f_i*(f_i - f_{i+1})*K) \quad .$$

This was generalized to

$$p_i = \exp(f_i^g*(f_i - f_{i+1})*K) \quad .$$

Although any $g \leq 0$ can be used, this investigation considers only $g = -1$. Standard simulated annealing is recovered from this generalization by setting $g = 0$ and using a predetermined set of values for K and a predetermined number of function evaluations at each K. Vanderbilt and Louie set $g = 0$ and use an indexed set of coefficients for K to force p_i to approach 0. In addition, they suggested a method for self-regulating the determination of the step size and the step distribution.

The GSA algorithm can be summarized using the following notation. Let F(x) be the real-valued function of interest evaluated at point $\mathbf{x}$, an element of some bounded subset of $\mathbf{R}^n$. Let Z be the global minimum value of F and let x_o be the initial set of independent variable values. The algorithm proceeds by:

1. Selecting x_o (randomly or based on other available information) and computing $F_o = F(x_o)$.

2. If this value is close enough to Z, stop; otherwise

3. Choose a direction from the uniform distribution on the unit hypersphere centered at x_o. Generate unit direction for U:

$$U_i = Y_i/(Y_1^2 + Y_2^2 + \ldots + Y_n^2)^{1/2}, \quad i = 1, \ldots, n$$

where Y_i is a standard normal deviate.

4. Choose a step size Δr and determine a new set of variable values
$$x = x_o + \Delta r*U \quad .$$

5. If x is not in the bounded support of F, generate a new x; otherwise,

6. If $F(x) < F_o$, accept x by setting $x_o = x$ and $F_o = F(x)$.

7. If $F(x) > F_o$, accept x with probability
$p = \exp(-Beta*(F(x)-F_o)/F_o)$
where Beta is a preset parameter. Otherwise, generate a new step.

8. Continue this random walk until $|F_o - Z| < \epsilon$, some arbitrary specified precision.

To use the algorithm to search for the optimum of a function, it remains to set the parameters r [the step size] and Beta [analogous to 1/(temperature of the system)]. A large Beta causes less movement than a small Beta. This is typically done by trial and error. The practical considerations are to

1. select Beta so that the

probability of accepting detrimental points is not too small (the algorithm can not escape local extrema) or too large (totally random walk);

2. select Δr so that the probability of exiting a local extremum is not too small (in which case the algorithm gets stuck too easily) or too large (leaves all extrema, including the global), given Beta.

In practice, the algorithm appears to perform best when about 60% (and between 50% and 90%) of the detrimental moves are accepted. The performance of the algorithm for a selected step size is influenced most by (1) the variability in the topography (values of F) over the support of F, (2) the range of the support, and (3) the number of dimensions. The next section illustrates this problem more specifically.

In addition to setting the parameters which govern the acceptance probability, a stopping rule must be specified. The most straightforward method, comparable to that used with several of the other stochastic optimization methods is to terminate the algorithm after a specified number of iterations without a move. The results reported in the next section are based on using 50 iterations without a move as a stopping rule; major shifts in this number, however, did not appear to have a large impact on the results.

COMPUTATIONAL RESULTS

Computational results applying a collection of stochastic optimization methods to a set of seven test functions were first collected in Dixon and Szego (1978), who proposed the standard test functions. Two other stochastic optimization methods were tested on the same set of functions by Rinnooy Kan and Timmer (1984, 1987); the specific coefficients for the test functions are given in Dixon and Szego. In this section, summary measures of the performance of the GSA algorithm are given for the same set of functions.

First a brief discussion of each problem is presented below followed by a summary list of the problems, and finally the solution results are presented.

The Goldstein-Price (GP) function is a two-dimensional function with three local and one global minima. An inverted view of the function is given in Figure 1. It shows the smoothness of the function along with the minima . Figure 2 shows the mean number of evaluations for various values of the parameter Beta and the step size. For any specified pair of parameter values, the variability in number of evaluations to termination is due to differences in the search path taken because of different random number seeds. This variability can be substantial in terms of number of evaluations, but for small problems such as this one the differences in CPU time are negligible.

The Branin (BR) function is a two dimensional function with three minima, all global. It is shown with an illustrative search path in Figure 3. Figure 4 shows the sensitivity of the mean number of evaluations to the two parameters for several representative values.

One function (H3) of the Hartman family is a three-dimensional function with five minima: four local minima and one global. The actual global is not the one reported in Dixon and Szego (1978); this function was difficult for GSA because the function is virtually flat in one dimension at the global, so the independent variable is unstable in that dimension prior to termination of the algorithm. The global found in this test was (approximately): (.11, .555,

Another function (H6) of the Hartman family is the six-dimensional version of H3. It was more stable at the global and, surprisingly, easier for GSA to terminate than the three-dimensional function. This held true for a wide range of parameter values and a large number of random seeds, although this performance does not seem to hold true for the other methods tested. Figure 5 summarizes the mean number of evaluations required on these test functions for some representative parameter values.

Three functions from the Shekel family - (S5), (S7), (S10) - were also tested. This series of functions in four dimensions has 5, 7, and 10 minima, respectively, each including one global minimum. This function family is the most difficult for the GSA method. A two-dimensional version shown in Figure 6 illustrates the reason: the depths of the local minima are great relative to the region of attraction at their mouths. The remainder of the surface is largely flat so that large step sizes tend to step over the regions of attraction and small step sizes fall in the local minimum they first encounter and are never able to escape.

The GSA algorithm was started from a number of boundary positions and one internal position and Figure 7 shows that the proportion of search paths terminating at each of the minima is proportional to the depth of the minimum, so that the largest proportion of searches terminates at the global. This is because the area of attraction for a minimum is proportional to its depth.

GENERAL SOLUTION METHOD

The precision of the solutions depends on the step size. The most efficient method for determining the global minimum of a function with appropriate precision was to first conduct a global search with a larger step size proportional to the volume of the bounded support for F in R^n. This phase proceeded by starting the GSA algorithm from several remote boundary positions and running 100 independent random search paths from each starting location (with reasonable parameter values determined by pre-sampling the function) to give some indication of variability in solution times and paths. The global phase located all the minima in all the test functions (except the shallowest minimum in the 10-minimum Shekel function). GSA always terminated at a minimum. Then the step size was adjusted for precision and a local search was conducted in the region of each of the minima found in the global search. To determine which local minimum is the global, a local search should be done in each region identified by the global searches. The GSA algorithm was run 100 times in each local region and found the value of the local minimum for that region for all runs on all functions. Using this general approach the GSA method found the global minimum to every test function to any arbitrary precision; the algorithm did not terminate at the global on every run for some functions, but multiple runs resulted in the highest proportion locating the global minimum for all the functions. Local searches always discriminated between local and global minima, and terminated in the local regions in which they began. This method also showed the approximate minimum previously given for H3 to be incorrect.

COMPUTATIONAL RESULTS

Table 1 gives a summary of the test functions described in this section and the parameter settings used to reach solutions. Table 2 lists the other global optimization methods used on these test problems.

The proportion of global searches terminating at the global is listed in the summary chart of computational results presented in Table 3. All other searches terminated at a local minimum.

Table 3 gives the number of function evaluations to termination for the various methods used on the test functions. Table 4 gives the same results in terms of standard time units where one unit is the CPU time to do 1000 evaluations of S5 at a specified location.

The results reported for GSA are the average time over 100 trials at the parameter values given in Table 2, starting from some remote boundary point in the support of F. The test results for the other methods are the averages of 4 independent runs and no variability measures or parameter settings are available.

What Tables 3 and 4 do not show is the sampling necessary to determine reasonable values for the parameters. This is not extensive, given initial settings related to the function properties, but is a component of the solution process (as it is for some of the other methods).

SUMMARY AND CONCLUSIONS

Computational experience with the generalized simulated annealing method for small problems over continuous variables indicates that the use of GSA may have some promise for problems of this type. The results are compared to a set of stochastic global optimization methods which represent some of the best alternatives available. GSA appears to be competitive in terms of solution times as well as reliability on this class of problems. The number of evaluations should be interpreted correctly: the number of local evaluations must be done for each of the minima located by the global search. The disadvantage of this procedure is that it requires a large amount of user interaction restarting the procedure at different points. The advantage of the procedure is that for small problems like these, this method provides a microcomputer-based ability to solve problems that we were not able to solve using Eureka, a commercially available micro-based steepest descent non-linear optimization package.

There are several potential modifications that could make the algorithm more efficient. Two current concerns are its sensitivity to the value of the parameters which govern the probability of acceptance (and to random seeds), and its lack of an operable stopping rule.

Results of using the algorithm on large-dimension continuous-variable problems (50 or more) have not been reported. Although it has proven successful on very large combinatorial problems, the behavior of the algorithm on these small problems with continuous variables indicates that there appear to be some serious potential problems for the algorithm to be a useful "general purpose" tool for solving very large problems (over 100 dimensions, say) with continuous variables. The method is sensitive to the parameter values determining the acceptance probability, the variability with respect to the random

seed is significant, for functions that are not very "smooth" it appears to be slow, and the two-phase procedure of global and local optimization requires substantial interaction on the part of the user.

REFERENCES

Bohachevsky, I.O., M. Johnson, M. Stein (1986). "Generalized Simulated Annealing for Function Optimization," Technometrics, 28, 209-217.

Branin, F., and S. Hoo (1972). "A Method for Finding Multiple Extrema of a Function of n Variables," in Numerical Methods of Nonlinear Optimization, ed. F. Lootsma, Academic Press, London.

Bremmerman, H. (1970). "A Method of Unconstrained Global Optimization," Mathematical Biosciences, 9, 1-15.

De Biasi, L., and F. Frontini (1978). "A Stochastic Method for Global Optimization: Its Structure and Numerical Performance," in Towards Global Optimization 2, ed. L. Dixon and G. Szego, North-Holland, Amsterdam.

Dixon, L., and G. Szego (1978). "The Global Optimization Problem," in Towards Global Optimization 2, ed L. Dixon and G. Szego, North-Holland, Amsterdam.

Geman,S., and D. Geman (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," IEEE Transactions Pattern Analysis and Machine Intelligence, PAMI-6, 721-741.

Golden, B.L. (to appear in 1988). American Journal of Mathematical and Management Sciences.

Hajek, B. (1986). "Optimization by Simulated Annealing: A Necessary and Sufficient Condition," in Adaptive Statistical Procedures and Related Topics; IMS Lecture Notes, ed. J. VanRyzin, Institute of Mathematical Statistics.

Kirkpatrick, S., D. Gelatt, and M. Vecchi, (1983). "Optimization by Simulated Annealing," Science, 220, 671-680.

Lundy, M. and S. Mees, (1986). "Convergence of an Annealing Algorithm," Mathematical Programming, 34, 111-124.

Price, W. (1978). "A Controlled Random Search Procedure for Global Optimization," in Towards Global Optimization 2, ed. L. Dixon and G. Szego, North-Holland, Amsterdam.

Rinnooy Kan, A., and G. Timmer, (1984). "Stochastic Methods for Global Optimization," American Journal of Mathematical and Management Sciences, 4, 7-40.

Rinnooy Kan, A., and G. Timmer, (1987). "Stochastic Global Optimization Methods Part II: Multilevel Methods," Mathematical Programming, 39, 57-78.

Torn, A. (1976). "Cluster Analysis Using Seed Points and Density Determined Hypersheres with an Application to Global Optimization," in Proceedings of the third International Conference on Pattern Recognition, Coronado, CA.

Vanderbilt, D., and S. Louie. (1984). "Monte Carlo Simulated Annealing Approach to Optimization Over Continuous Variables," Journal of Computational Physics, 56, 259-271.

Table 1: Test Functions and Values of Beta and Δr to Solve Them.

Name	Dim	Minima	Form	Beta	Δr
Goldstein-Price (GP)	2	4	Eighth-order polynomial	G: 1 L: 100	.2 .005
Branin (BR)	2	4	Fourth-order polynomial plus cosine term	G: 10 L: 500	.75 .02
Hartman (H3)	3	4	$-\Sigma c_i \exp\{-x'Ax\}$	G: 50 L: 2500	.05 .005
Hartman (H6)	6	4	(same)	G: 45 L: 2500	.07 .005
Shekel 5 (S5)	4	5	$-\Sigma\{\|x-p_i\|^2 + c_i\}^{-1}$	G: 50 L: 500	1.75 .01
Shekel 7 (S7)	4	7	(same)	G: 50 L: 500	1.75 .01
[illegible]	[illegible]	[illegible]	(same)	G: 50 L: 500	1.75 .01

Dim: Number of dimensions
G: Global search
L: Local search

Table 2: Optimization Methods used on Test Problems

Method	Description
Trajectory (T)	Gradient path method (Branin and Hoo (1972))
Random direction (RD)	Random directions (Bremmerman (1970))
Controlled random search (CR)	Price (1978)
Density clustering (DC)	Sample concentration and clustering (Torn(1976))
Density reduction (DR)	Density clustering, reduction and spline fitting (De Biasi and Frontini (1978))
Multi-level single linkage (ML)	Clustering by distance (Rinnooy Kan and Timmer (1987))
Self-regulating (SRSA)	Annealing with self-adjusting step determination (Vanderbilt and Louie (1984))
Generalized simulated annealing (GSA)	Section 2, this article

Table 3: Number of Function Evaluations to Find the Global Minimum

Function	Method T	RD	CR	DC	DR	ML	SRSA / P	GSA** / P
GP	-	300	2500	2499	378	294	1186 / 99	170 + 122 / 100
BR	-	160	1800	1558	597	219	557 / 100	121 + 115 / 100
H3	-	*	2400	2584	732	370	1224 / 100	310 + 145 / 78
H6	-	515	7600	3447	807	877	1914 / 62	287 + 235 / 100
S5	5500	*	3800	3649	620	347	3910 / 54	400 + 296 / 58
S7	5020	*	4900	3606	788	399	3421 / 64	261 + 296 / 47
S10	4860	*	4400	3874	1160	447	3078 / 81	224 + 296 / 47

/P: Proportion of trials ending at the global; remainder at locals
-: No results available
*: Failed to find global
**: Number of global evaluations plus local evaluations

Table 4: Number of Standard Time Units to Find the Global Minimum

Function	Method T	RD	CR	DC	DR	ML	SRSA	GSA**
GP	-	0.7	3	4	15	0.4	2	.6
BR	-	0.5	4	4	14	0.4	1	.4
H3	-	*	8	8	16	1	4	1.2
H6	-	3	46	16	21	3	12	2.0
S5	9	*	14	10	23	.75	16	2.1
S7	8.5	*	20	13	20	1	15	1.7
S10	9.5	*	20	15	30	1.5	15	1.9

Standard time unit: CPU time for 1000 function evaluations of S5.

-: No results available
*: Failed to find global
**: Number of global evaluations plus local evaluations

Figure 1. GP Function (inverted)

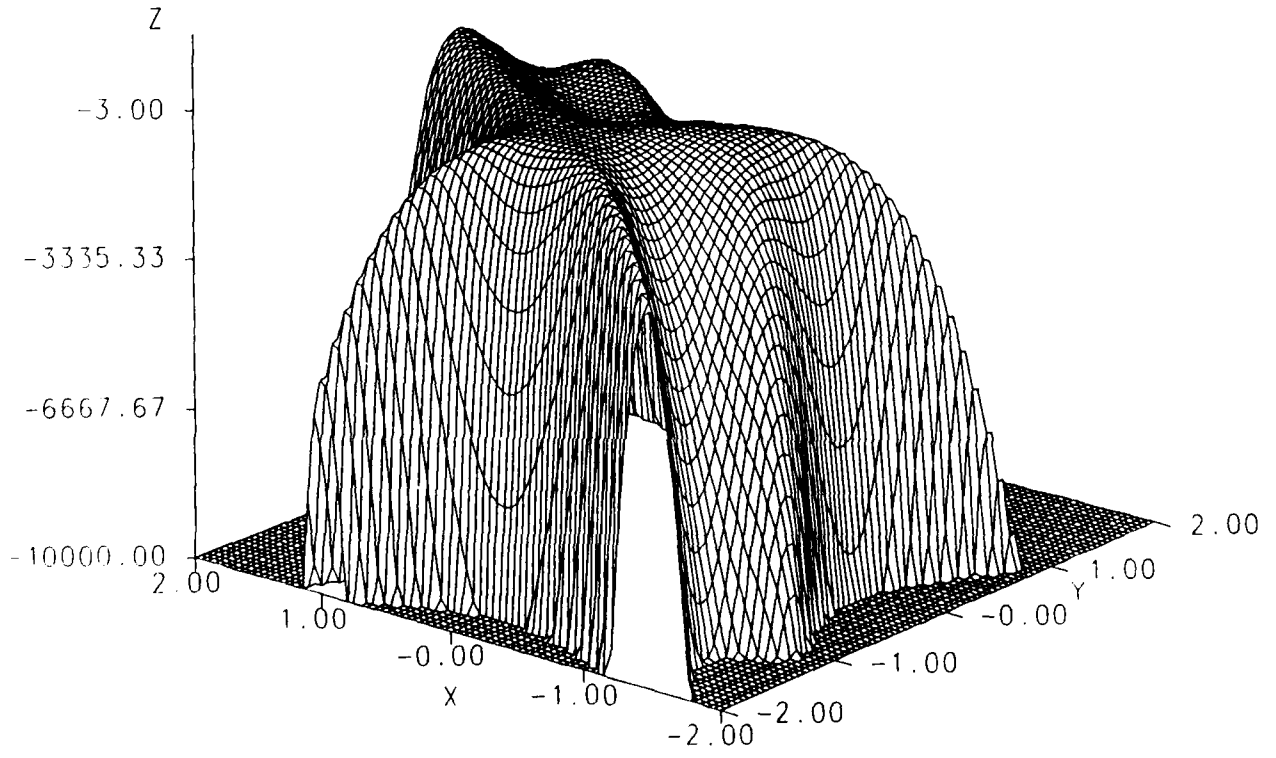


Figure 2. Goldstein–Price

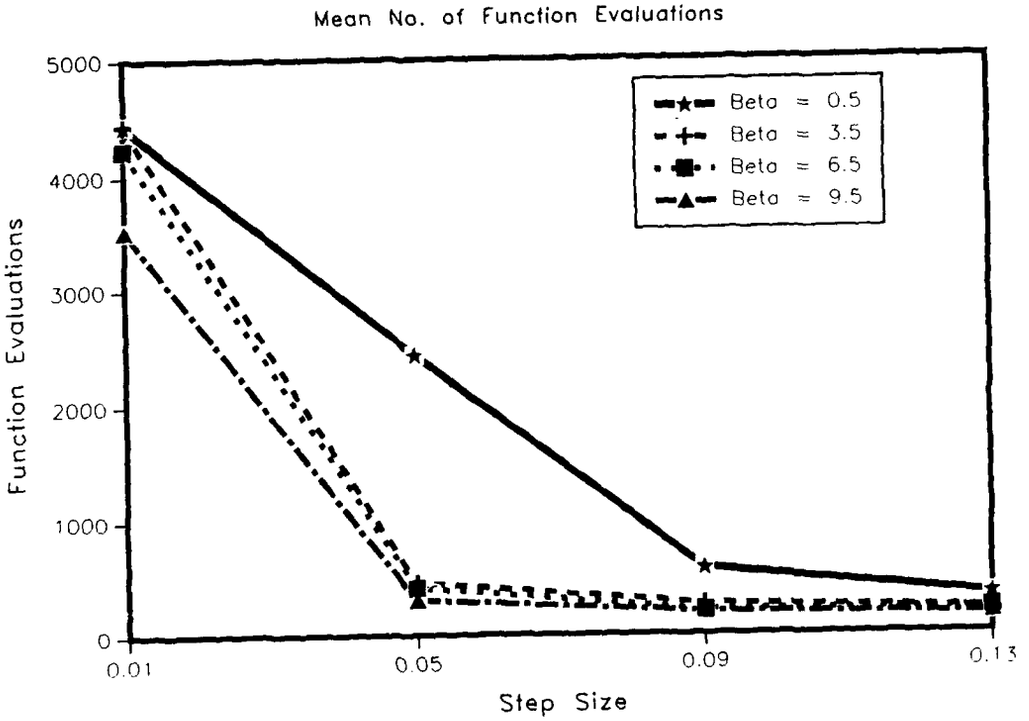


Figure 3. BR Function

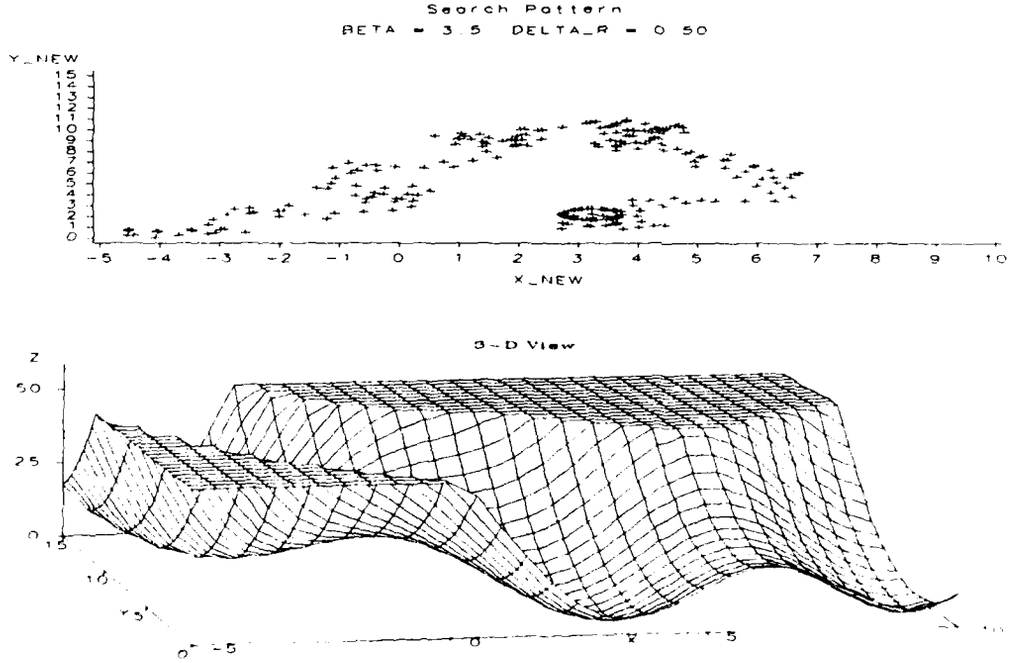


Figure 4. Branin

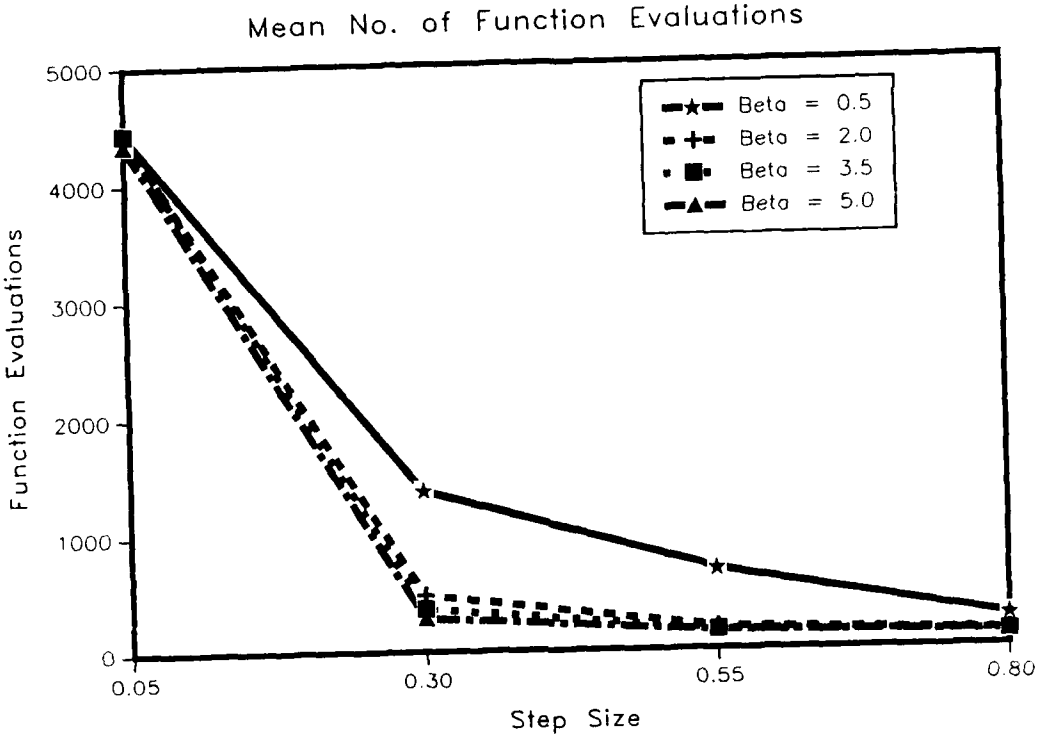


Figure 5

Mean No. of Function Evaluations

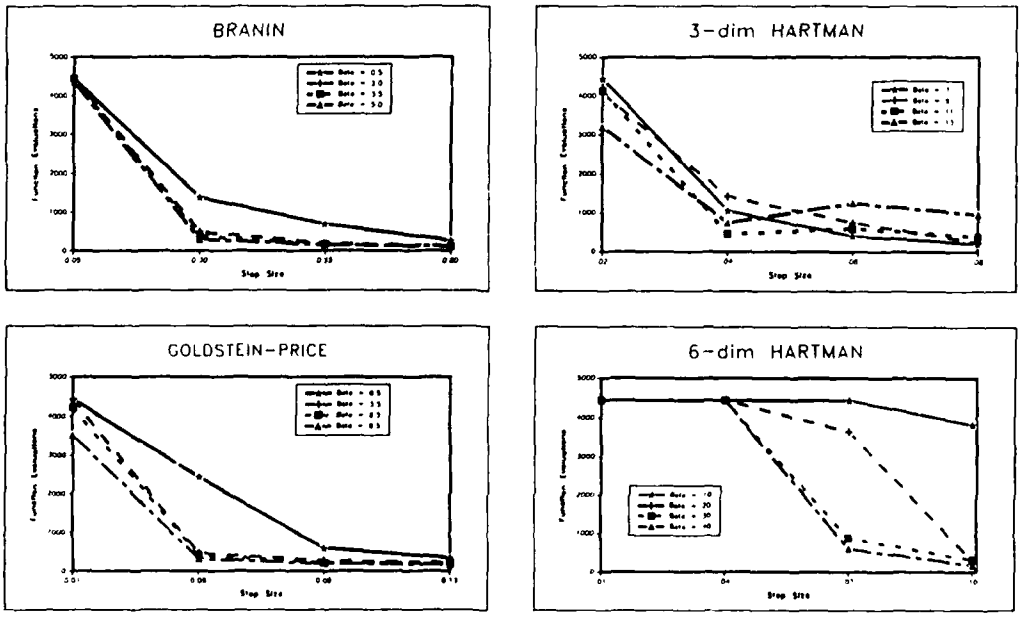


Figure 6. 2-D Shekel

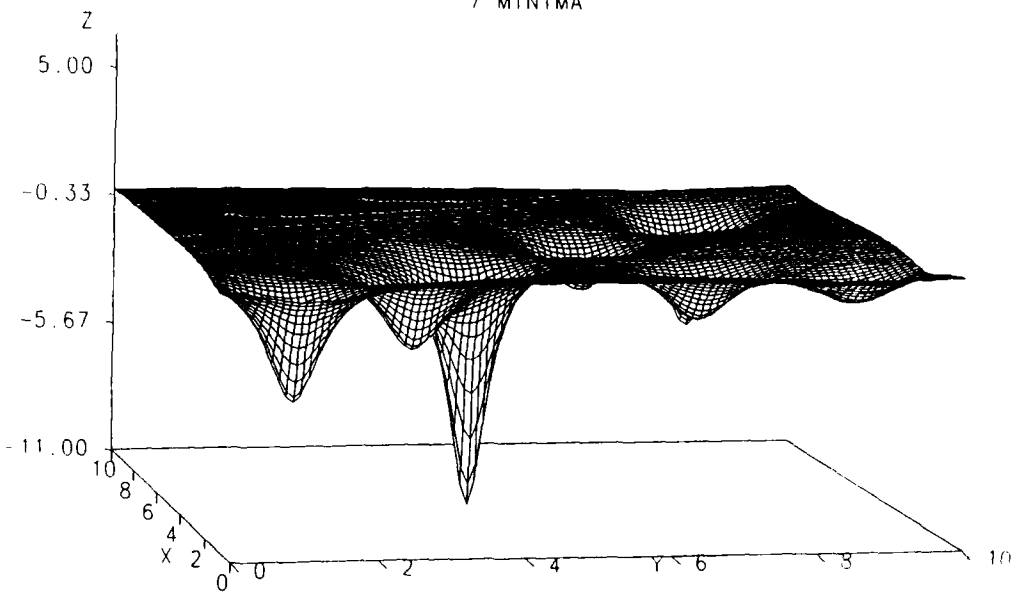
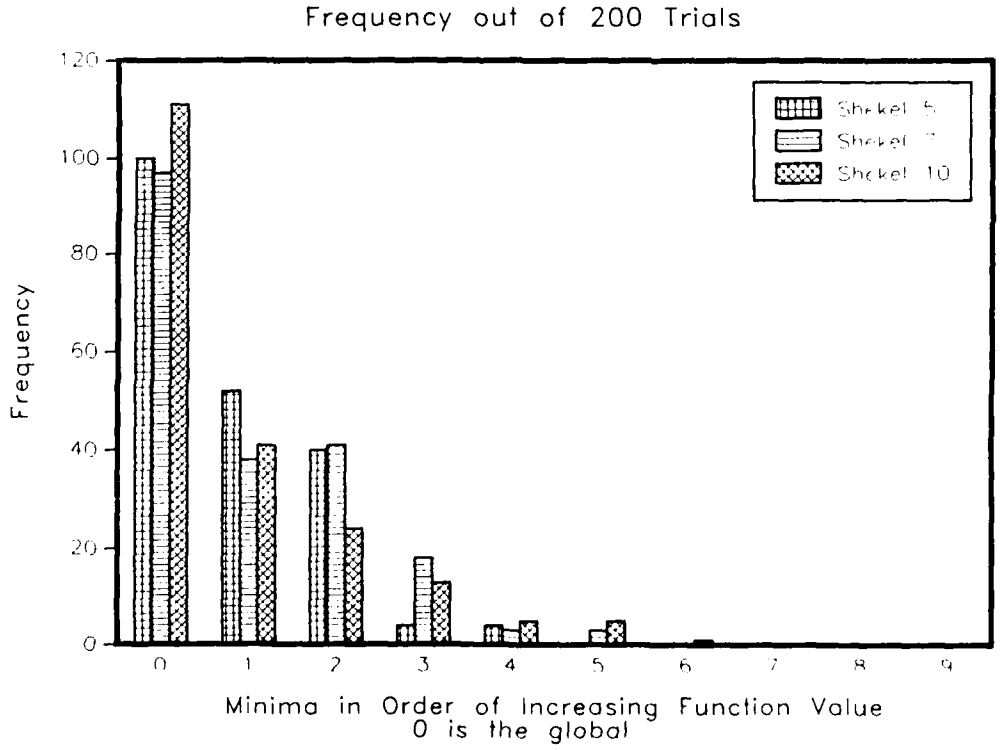


Figure 7. Where Search Terminated



SIMULATED ANNEALING IN THE CONSTRUCTION OF EXACT OPTIMAL DESIGNS

Ruth K. Meyer, St. Cloud State University
Christopher J. Nachtsheim, University of Minnesota

Introduction

Exact optimal design of experiments is concerned with specifying n points from a design space at which observations are to be taken in order to achieve precise estimation. A linear model of the form

$$Y = X\beta + \varepsilon$$

is assumed, where Y is an nx1 vector of observations, X is the nxp design matrix, β is a px1 vector of unknown regression parameters, and ε is an nx1 vector of uncorrelated experimental errors with mean zero and constant variance σ^2. The ith observation y_i is obtained at a vector-valued point x_i in a q-dimensional compact design space χ, and the corresponding row of X is written $f'(x_i)$. For example, consider a second order response surface model with two factors,

$$f'(x_i) = (1, x_{i1}, x_{i2}, x_{i1}^2, x_{i2}^2, x_{i1}x_{i2}).$$

If the parameters are estimated by least squares, the variance of the estimate of β is given by $\sigma^2(X'X)^{-1}$. The variance of the fitted values at x_i is proportional to

$$d(x_i) = f'(x_i)(X'X)^{-1}f(x_i),$$

termed the variance function.

Designs are chosen using one or more optimality criteria. Generally such criteria are represented by functionals on the pxp covariance matrix $(X'X)^{-1}\sigma^2$ (See Steinberg and Hunter, 1984, for a review). The most widely applied criterion is D-optimality, first proposed by Wald (1943). D-optimal designs maximize $|X'X|$, in effect, minimizing the generalized variance of the estimated coefficients. If the errors (ε_i) are normally distributed, the design minimizes the volume of a fixed level confidence ellipsoid for β.

If X* is the design matrix corresponding to the D-optimal design, the D-efficiency of any other n-point design is given by $100(|X'X|/|X^{*\prime}X^{*}|)^{1/p}$. If the D-optimal design is unknown, as is often the case, the relative efficiency,

$$\text{R-efficiency} = 100(|X_1'X_1|/|X_2'X_2|)^{1/p},$$

is typically used to compare n-point designs having respective design matrices X_1 and X_2.

Early efforts in D-optimal design construction used mathematical programming techniques to directly maximize $|X'X|$ (See e.g.,Box, 1966). Box and Draper (1971) used Powell's direct search to maximize $|X'X|$ in up to 30-dimensional space. More recently various exchange algorithms, for example, Mitchell's DETMAX (1974), Federov (1972), k-exchange (Johnson and Nachtsheim, 1983), reduce the dimension of the search space. These algorithms begin with a nonsingular n-point design and iteratively add a point from the design space and delete a point from the current design such that a maximal increase in $|X'X|$ is obtained. The exchange of design points typically is determined by computing optima of the variance function, deleting the point with minimum variance of prediction and adding the point with maximum variance. Convergence of the sequence, however, may be to a locally optimal design.

When χ is finite, various simplifications result. For example, optimization of the variance function can be globally obtained at each iteration. Moreover, when there are N design or "candidate" points in χ, there are $\binom{n+N-1}{n}$ possible designs (Welch, 1982), making an exhaustive search theoretically possible. Welch (1982) developed a branch-and-bound algorithm which guarantees global exact D-optimal designs, but is computationally infeasible with large dimensional problems.

However, design spaces are often represented by convex regions in R^q, and the simplifications described above are not applicable. Cook and Nachtsheim (1980) and Johnson and Nachtsheim (1983) have advocated the use of exchange algorithms with embedded nonlinear optimization routines to determine the points to exchange. Cook and Nachtsheim (1980) used a combined grid-Powell search in an attempt to locate the D-optimal design. Meyer and Nachtsheim (1987) implemented GRG2, a generalized reduced gradient method for nonlinear optimization, within the k-exchange algorithm.

One inherent difficulty associated with the use of nonlinear optimization routines is the convergence at local optima. As the dimension of the problem and the number of terms in the model increase, the number of local optima of the variance function increases. In an attempt to surmount the obstacles encountered with current algorithms, we implement the simulated annealing algorithm to directly maximize the determinant of X'X, and evaluate its performance on both finite and convex design spaces.

Haines (1987) applied the simulated annealing algorithm to construct various n-point optimal designs using several criteria for polynomial regression of up to degree 5 and for the second order model with 2 factors. Trial designs were constructed by successively perturbing individual points. The algorithm was most effective in constructing G-optimal designs that minimize the maximum variance function.

We modify the generalized simulated annealing method described by Bohachevsky, Johnson, and Stein (1986) to maximize $|X'X|$. This algorithm, which has the "ability to migrate through a sequence of local extrema in search of the global solution and to recognize when the global extremum has been located" (Bohachevsky, et al. 1986, p. 209) substantially improved the D-optimal 11-point design for a specific nonlinear problem with many constraints given by Bates (1983). Generalized simulated annealing makes the probability of accepting a detrimental step

tend to zero as the random walk approaches the global optimum.

Application of the Generalized Simulated Annealing Algorithm

The models we consider are the first order and second order response surface models. The first order model

$E(y) = \beta_0 + \Sigma\beta_j x_j, \quad j = 1,2,\ldots,q$ has $p = q + 1$ parameters. The second order model

$E(y) = \beta_0 + \Sigma\beta_j x_j + \Sigma\Sigma_{j\leq k}\beta_{jk} x_j x_k, \quad j,k = 1,2,\ldots,q$

contains $p = (q+1)(q+2)/2$ parameters.

The construction of D-optimal designs requires the maximization of $|X'X|$, the value of which is quite large particularly for problems with many factors or many design points. The values of the parameters used in applying the generalized simulated annealing method are simpler to adjust if the objective function is defined as the maximization of $|X'X|^{1/p}$, where p is the number of parameters in the model. To ensure the desired behavior of the probability near the global optimum, the objective function is defined to converge to zero at the global optimum. The maximization of $|X'X|^{1/p}$ is thus substituted by its equivalent, the minimization of $\phi(X) = D_{max} - |X'X|^{1/p}$, where D_{max} is the user's prior estimate of the optimum determinant. If the value of the maximum determinant is understated and $\phi(X)$ becomes negative, D_{max} is increased and the search is continued. If the estimate is too large, D_{max} is decreased to ensure the objective function converges to zero.

Finite Design Spaces

The D-optimal design for first order models has been shown to consist entirely of the vertices of the q-dimensional hypercube (Box and Draper, 1971 and Mitchell, 1974). The candidate set contains 2^q points; each coordinate of x_i is -1 or +1. For second order models we assume each coordinate may be -1, 0 or +1, defining 3^q points in the candidate set.

The algorithm begins with an nxq starting matrix consisting of the coordinates of n points chosen randomly from the candidate set. At each iteration, a trial design is defined by perturbing $m < nq$ of the coordinates. If the value of the objective function is decreased, the trial design is accepted with probability 1. If the value is increased, the trial design is accepted with probability

$$p = \exp\{-\theta\Delta\phi(X)/\phi(X_0)\}$$

where θ is a nonnegative control parameter, $\Delta\phi(X)$ is the change in the objective function value, and $\phi(X_0)$ is the current value of the objective function. The appropriate values for m and θ depend on particular problem characteristics and are found by experimentation. As the algorithm is executed, the value of m is gradually decreased to a minimum of 1 as the global minimum is approached, in which case a single coordinate change is made to define a trial design.

The steps of the algorithm with finite design spaces are as follows:

1. Generate a random starting design, X_0.
2. Calculate $\phi(X_0)$. If $|\phi(X_0)| \leq \epsilon$, go to 7.
3. Determine a trial design, X, by randomly selecting m coodinates to change.
   a. For first order models:
      If $x_{ij} = -1$, set $x_{ij} = +1$.
      If $x_{ij} = +1$, set $x_{ij} = -1$
   b. For second order models:
      If $x_{ij} = -1$, set $x_{ij} = 0$.
      If $x_{ij} = +1$, set $x_{ij} = 0$.
      If $x_{ij} = 0$, set $x_{ij} = +1$ with probability .5.
      Set $x_{ij} = -1$ with probability .5.
4. Calculate the new value of the objective function $\phi(X)$ and let
   $\Delta\phi(X) = \phi(X_0) - \phi(X)$.
   If $|\phi(X)| \leq \epsilon$, go to 7.
5. If $\phi(X) \leq \phi(X_0)$, let $X_0 = X$ and $\phi(X_0) = \phi(X)$. Go to 3.
6. If $\phi(X) > \phi(X_0)$,
   let $p = \exp\{-\theta\Delta\phi(X)/\phi(X_0)\}$ and generate a uniform [0,1] random variable, u.
   a. If $u \geq p$, go to 3.
   b. If $u < p$, let $X_0 = X$ and $\phi(X_0) = \phi(X)$. Go to 3.
7. Stop.

Convex Design Spaces

We consider the design space most often used in experimentation, the q-dimensional hypercube defined by

$-1 \leq x_{ij} \leq 1; \quad i = 1,2,\ldots,n, \quad j = 1,2,\ldots,q.$

Since many of the optimal design points occur at the vertices of the design space, the algorithm performed better if the constraint set on the design space was eliminated. A useful transformation described by Box (1966) and used by Atkinson (1969) for D-optimal design computations is

$x_{ij} = \sin y_{ij}$, for $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,q$.

Then for all values of y_{ij}, $-1 \leq x_{ij} \leq 1$.

A trial design matrix is determined by perturbing each transformed coordinate by the amount $\Delta m * v_{ij}$, where v_{ij} is a random direction in nq-dimensional space and Δm is the step size. The trial design is accepted with probability 1 if the value of the objective function is decreased, and accepted with probability $p = \exp\{-\theta\Delta\phi(X)/\phi(X_0)\}$ if the value is increased. The values selected for Δm and θ depend on particular problem characteristics and are found by experimentation. The value of Δm is decreased gradually during execution of the algorithm to refine the design as the global minimum is approached.

The algorithm for a convex design space follows:

1. Generate a random starting design, X_0.
2. Calculate $\phi(X_0)$. If $\phi(X_0)$, go to 9.
3. Let $Y_0 = \arcsin X_0$.

4. Determine an nxq random direction matrix V by choosing independent uniform [-1,1] random variables, b_{ij}, and computing the components of V: $v_{ij} = b_{ij}/(\sum b_{ij})^{1/2}$.
5. Let $Y = Y_0 + \Delta mV$; let $X = \sin Y$.
6. Calculate the new value of the objective function $\phi(X)$; let $\Delta\phi(X) = \phi(X_0) - \phi(X)$. If $|\phi(X)| \leq \epsilon$, go to 9.
7. If $\phi(X) \leq \phi(X_0)$, let $X_0 = X$ and $\phi(X_0) = \phi(X)$. Go to 3.
8. If $\phi(X) > \phi(X_0)$, let $p = \exp\{-\Delta\phi(X)/\phi(X_0)\}$ and generate a uniform [0,1] random variable, u.
   a. If $u \geq p$, go to 4.
   b. If $u < p$, let $X_0 = X$ and $\phi(X_0) = \phi(X)$. Go to 3.
9. Stop.

Results

The algorithms were executed on the Cray-2 supercomputer at the University of Minnesota using test problems for first and second order response surface models on both finite and convex design spaces. A detailed account of the empirical results is contained in Meyer and Nachtsheim (1988).

Conclusions

The generalized simulated annealing algorithm was used to construct D-optimal designs on both finite and convex design spaces in an attempt to overcome the problems of premature convergence and/or computer infeasibility with high dimensions encountered with current algorithms. For the finite design space, the only algorithm currently available for construction of globally optimal designs is Welch's (1982) branch-and-bound search, which is not recommended if N > 30. Our results suggest that the generalized simulated annealing algorithm can be simply implemented and cheaply used to search for globally optimal designs on as many as N = 1000 candidate points. We have demonstrated its utility for first order response surface models having up to 10 factors, and for second order models with as many as 5 factors. The cost, however, is that D-optimality is not guaranteed.

Conversely, our results are not encouraging in the presence of convex design spaces. Considerably more computer time was required for the construction of D-optimal designs on convex spaces than on finite spaces. The convergence rate was slower, requiring many more determinant evaluations. As Haines (1987) did, a more sensitive search method, such as Fiacco's (1964) conjugate direction method, is needed to refine the design. For higher dimensional problems on convex spaces, the generalized simulated annealing method may have its greatest potential embedded within an exchange algorithm as an added check for convergence.

Acknowledgement

This research was funded in part by the Minnesota Supercomputer Institute of the University of Minnesota.

References

Atkinson, A. C. (1969) "Constrained Maximization and the Design of Experiments" TECHNOMETRICS, 11, 616-618.

Bohachevsky, Ihor O., Mark E. Johnson, and Myron L. Stein (1986) "Generalized Simulated Annealing for Function Optimization" TECHNOMETRICS, 28, 209-217.

Box, M. J. (1966) "A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems" COMPUTER JOURNAL, 9, 67-77.

Box, M. J. and Draper, N. R. (1971) "Factorial Designs. the X'X Criterion, and Some Related Matters" TECHNOMETRICS, 13, 731-742.

Cook, R. Dennis and Christopher J. Nachtsheim (1980) "A Comparison of Algorithms for Constructing Exact D-Optimal Designs" TECHNOMETRICS, 22, 315-324.

Federov, V.V. (1972) THEORY OF OPTIMAL EXPERIMENTS, New York Academic Press.

Galil, Z., and Kiefer, J. (1980) "Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-optimum Designs" TECHNOMETRICS, 22, 301-313.

Haines, Linda M. (1987) "The Application of the Annealing Algorithm to the Construction of Exact Optimal Designs for Linear-Regression Models" TECHNOMETRICS, 29, 439-447.

Johnson, Mark E. and Christopher J. Nachtsheim (1983) "Some Guidelines for Constructing Exact D-Optimal Designs on Convex Design Spaces" TECHNOMETRICS, 25, 271-277.

Meyer, Ruth K. and Christopher J. Nachtsheim (1987) "Optimal Design of Experiments in the Presence of Irregularly Constrained Design Regions" Paper presented at the American Statistical Association Annual Meeting, San Francisco.

Meyer, Ruth K. and Christopher J. Nachtsheim (1988) "Simulated Annealing in the Construction of Exact Optimal Design of Experiments" Technical Report 88/48, University of Minnesota Supercomputer Institute.

Mitchell, T.J. (1974) "An Algorithm for the Construction of 'D-Optimal Experimental Designs'" TECHNOMETRICS, 16, 203-210.

Mitchell, T.J. (1974) "Computer Construction of "D-Optimal" First-Order Designs" TECHNOMETRICS, 16, 203-210.

Steinberg, David M. and William G. Hunter (1984) "Experimental Design: Review and Comment" TECHNOMETRICS, 26, 71-97.

Wald, A. (1943) "On the Efficient Design of Statistical Investigations," ANNALS OF MATHEMATICAL STATISTICS, 14, 134-140.

Welch, William J. (1982) "Branch-and-Bound Search for Experimental Designs Based on D Optimality and Other Criteria" TECHNOMETRICS, 24, 217-224.

A SIMULATED ANNEALING APPROACH TO MAPPING DNA

LARRY GOLDSTEIN MICHAEL S. WATERMAN

UNIVERSITY OF SOUTHERN CALIFORNIA

Summary

The double digest mapping problem that arises in molecular biology is an NP complete problem that shares similarity with both the travelling salesman problem and the partition problem. Sequences of DNA are cut at short specific patterns by one of two restriction enzymes singly and then by both in combination. From the set of resulting lengths, one is required to construct a map showing the location of cleavage sites. In order to implement the simulated annealing algorithm, one must define appropriate neighborhoods on the configuration space, in this case a pair of permutations, and an energy function to minimize that attains its global minimum value at the true solution. We study the performance of the simulated annealing algorithm for the double digest problem with a particular energy function and a neighborhood structure based on a deterministic procedure for the travelling salesman problem.

1 Introduction

The simulated annealing algorithm has shown promise on a variety of combinatorially hard problems, such as the NP complete travelling salesman problem [1]. Below, we study the performance of an implementation of the simulated annealing algorithm on the double digest mapping problem, an NP complete problem arising in molecular biology. The double digest mapping problem can be stated roughly as follows. A restriction enzyme cuts a strand of DNA, regarded as a finite sequence over the four letter alphabet $\{A,C,G,T\}$, at all occurrences of a pattern specific to that enzyme; the patterns are short, typically 4 to 6 letters in length. Only the resulting fragment lengths are recorded. When two enzymes are used singly and then in combination, it is required to construct a map based on these three sets of recorded lengths, showing the location of all cleavage sites. For previous work in this area see [13], [12], [3], [11], [2], and [14].

It is perhaps not surprising that the double digest problem in a member of the class of NP complete problems, a class of problems for which no polynomial time algorithms are known. This may be demonstrated by showing that a special case of the double digest problem is an NP complete problem known as the partition problem. Hence, the double digest problem is at least as hard as the partition problem, and itself belongs to the class of NP hard problems.

Given that therefore it is unlikely one will find a fast, polynomial time algorithm to solve the double digest problem, one may turn to the simulated annealing algorithm, a recent probabilistic procedure that has enjoyed some success on combinatorially hard problems of this nature.

This paper is a report on the application of the simulated annealing algorithm to the double digest problem. We first give a mathematical description of the double digest problem. Next, we show that the double digest problem is an NP complete problem. In the section that follows, we give a description of the simulated annealing algorithm in general, and state how it may be applied to the problem at hand. Lastly, we conclude with some remarks on the effectiveness of the procudere in this instance and on the nonuniqueness of solutions to the double digest problem in general.

2 Description of the Double Digest problem

The double digest problem can be stated as follows. A restriction enzyme cuts a piece of DNA of length L at all occurrences of a short specific pattern and the lengths of the resulting fragments are recorded. In the double digest problem we have as data the list of fragment lengths when each enzyme is used singly, say

$$A = \{a_1, \ldots, a_m\}$$

$$B = \{b_1, \ldots, b_n\}$$

from digest A,

$$B = \{b_i : 1 \le i \le m\}$$

from the second digest, as well as a list of double digest fragment lengths when the restriction enzymes are used in combination and the DNA cut at all occurrences specific to both patterns, say

$$C = \{c_i : 1 \le i \le n_{1,2}\};$$

only length information is retained. In general A,B and C will be multisets; that is, there may be values of fragment lengths that occur more than once. We adopt the convention that the sets A,B,and C are ordered, that is, $a_i \le a_j$ for $i \le j$, and likewise for the sets B and C. Of course

$$\sum_{1 \le i \le n} a_i = \sum_{1 \le i \le m} b_i = \sum_{1 \le i \le n_{1,2}} c_i = L,$$

since we are assuming that fragment lengths are measured in number of letters with no errors. Given the above data the problem is to find orderings for the sets A and B such that the double digest implied by these orderings is, in a sense made precise below, C.

We may express the double digest problem more precisely as follows. Let S_k denote the set of all permutations on k objects. For $\sigma \in S_n, \mu \in S_m$ call (σ, μ) a configuration. By ordering A and B according to σ and μ respectively, we obtain the set of locations of cut sites

$$S = \{s : s = \sum_{1 \le j \le r} a_{\sigma j} \text{ or}$$

$$s = \sum_{1 \le j \le t} b_{\mu j} ; 0 \le r \le n, 0 \le t \le m\}.$$

Since we want to record only the location of cut sites, the set S is not allowed repetitions, that is, S is not a multiset. Now label the elements of S such that

$$S = \{s_j : 0 \le j \le n_{1,2}\}$$

with $s_i \le s_j$ for $i \le j$.

The double digest implied by the configuration (σ, μ) can now be defined as the lengths that result when the fragment is cut at the locations indicated by S, that is, by

$$C(\sigma, \mu) = \{c_i(\sigma, \mu) : c_i(\sigma, \mu) = s_j - s_{j-1}$$

$$\text{for some } 1 \le j \le n_{1,2}\}$$

where we assume as usual that the set is ordered in the index i. The problem then is to find a configuration (σ, μ) such that $C = C(\sigma, \mu)$.

We note for future reference that the function f on the configuration space given by

$$f(\sigma, \mu) = \sum_{1 \le i \le n_{1,2}} (c_i(\sigma, \mu) - c_i)^2 / c_i$$

attains its global minimum value of zero at the configuration (σ, μ) if and only if this configuration is a solution to the double digest problem. Hence, we may consider an equivalent formulation of the double digest problem: find where f attains its global minimum value of zero.

3 Computational Complexity of the Double Digest Problem

We demonstrate below that the double digest problem is NP complete. It is clear that the double digest problem DDP as described above is in the class NP, as a nondeterministic algorithm need only guess a configuration (σ, μ) and check in polynomial time if $C(\sigma, \mu) = C$. The number of steps to check this is in fact linear. To show that DDP is NP complete we transform the partition problem to DDP. In the partition problem, known to be NP complete [4], we are given a finite set A, say $|A| = n$, and a positive integer $s(a)$ for each $a \in A$ and wish to determine whether there exists a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a).$$

If $\sum_{a \in A} s(a) = J$ is not divisible by two, there can be no such subset A'; else, consider as input to problem DDP the data

$$A = \{s(a_k) : 1 \leq k \leq n\}$$

$$B = \{J/2, J/2\} \quad \text{and set } C = A.$$

It is clear that any solution to problem DDP with this data yields a solution to the partition problem through the order of the implied digest C.

4 The Simulated Annealing Algorithm

We now give a description of the simulated annealing algorithm. The algorithm is based on the following analogy with statistical mechanics. To a given physical system, there corresponds a function f that assigns to the state of that system its energy. The algorithm mimics the behavior of such a physical system moving from state to state in order to minimize energy.

Specificaly, let V be a finite set of elements, and f a function that assigns a real number to each element of V. The elements of V represent the state of the system, and we think of $f(v)$ as the energy of the system when in state v.

In statistical mechanics, the Gibbs distribution gives the probability of finding the system in a particular state. Introducing the temperature parameter T, we write the Gibbs distribution as

$$\pi_T(v) = exp\{-f(v)/T\}/Z_T,$$

where Z_T, the partition function, is chosen such that

$$\sum_{v \in V} \pi_T(v) = 1.$$

For large values of T the distribution tends to be uniform over V, while for small values of T the favorable elements of V, that is, those elements of V for which $f(v)$ is small, are weighted with large probability. Therefore, a probabilistic solution to the problem of locating an element $v \in V$ for which $f(v)$ is minimized is given by sampling from the distribution π_T for small $T > 0$.

One way this may be achieved is to simulate a Markov chain $\{X_n\}_{n \geq 0}$ with state space V that has π_T as its stationary distribution and let it approach equilibrium. It is possible to write down an explicit formula for the transition law of such a Markov chain.

Simulating a Markov chain of this type with the parameter T fixed was proposed by Metropolis et al. [10]. One may observe that in the context of minimization, the smaller the value of T the higher the probability of finding the state of global minimum energy. Kirkpatrick et al. [8] introduced the idea of cooling the system in the hope that in the limit one would obtain the distribution π_0 that puts mass one uniformly over the states of minimum energy. In this way the algorithm resembles the physical process of annealing, or cooling, a physical system. As in the physical analog, the system may be cooled too rapidly and become trapped in a state corresponding to a local energy minimum; Geman and Geman [6] (see also Hajek [7]) showed that if at stage n in the algorithm one cools the system with a sequence of temperatures T_n,where $T_n \downarrow 0$ and $T_n \geq c/\log(n)$ with c a constant that depends on f, then the state of the Markov chain converges in distribution to π_0.

In order to simulate the Markov chain it is required to specify, for each possible state v, the collection of states N_v where transitions are to be allowed. We call such a collection of states *neighbors* of a given state. Of course, we must require each state to be reachable from any other state through a sequence of neighbors.

Our neighborhood structure was motivated by a neighborhood structure used in a simulated annealing algorighm for the travelling salesman problem [1], which in turn was based on a deterministic procedure for that particular problem [9]. In the travelling salesman problem one is required to find the tour of shortest length that

visits n given cities in the plane. Hence, the configuration space for the travelling salesman problem is the set of permutations, where a particular permutation gives the order in which cities are to be visited. For the double digest problem, as described in section 2, a configuration is a pair of permutations.

We now describe a neighborhood structure for the travelling salesman problem([1], [9]). If, for a given permutation, or tour, σ we imagine links connecting cities in the tour, we say that the tour σ is k-optimal, or k-opt for $1 \leq k \leq n$, if for all tours that can be obtained from σ by breaking at most k links, the tour given by σ is the shortest. Thus, every tour is 1-opt and only the true best tours are n-opt. We define a neighborhood system using the concept of 2-optimality. For a given tour $\sigma = (i_1, i_2, .., i_n)$ visiting city i_1 then i_2 and so on, let the neighborhood of σ be defined by

$$N(\sigma) = \{\tau \in S_n : \tau = (i_1, i_2, .., i_{j-1}, i_k, i_{k-1}, .., i_{j+1}, i_j, i_{k+1}, ..., i_n) \text{ for some } 1 \leq j \leq k \leq n\}.$$

It is not difficult to see that this notion of neighborhood allows one to transition from any state to any other state through a sequence of neighborhs.

For the double digest problem our configuration space is a pair of permutations. Accordingly, for this problem we may define a neighborhood of a configuration (σ, μ) by

$$N(\sigma, \mu) = \{(\tau, \mu) : \tau \in N(\sigma)\} \cup \{(\sigma, \nu) : \nu \in N(\mu)\}$$

where $N(\rho)$ are the neighborhoods used in the discussion of the travelling salesman problem above.

We conclude this section with an explicit description of the simulated annealing algorithm.

Let the initial state v_1 be an arbitrary element of the configuation space S. At stage n, let us say the state of the system is $u = (\sigma, \mu)$. Set $T_n = \frac{c}{\log n}$. Select a neighbor $v \in N_u$ uniformly from N_u. For the case at hand, this selection may be done in the following manner. Choose to invert either σ or τ, each with equal probability. Say σ is chosen. We now randomly invert a portion of the "tour" given by σ in such a way that all inversions are equally likely, yielding a new "tour", say τ. Let $v = (\tau, \mu)$. Compute $\Delta = f(v) - f(u)$. If $\Delta < 0$ then accept v as the new state of the chain for iteration $n + 1$. If $\Delta \geq 0$, accept v as the new state of the chain with probability $p = \exp\{-\Delta/T_n\}$ and keep u as the new state for iteration $n + 1$ with probability $1 - p$.

5 Performance of the Algorithm

With the above framework in place, the simulated annealing algorithm was run on both simulated and actual mapping problems.

The performance of the algorithm on large simulated problems led us to suspect that in general solutions to the double digest mapping problem are not unique. In fact, under a certain probability model, the number of solutions to the double digest mapping problem increases exponentially in the length of the segment [5]. The performance of the algorithm for these problems is therefore confounded by the large number of exact solutions.

For mapping the bacteriophage lambda, 45,360 base pairs in length, with the restriction enzymes BamH1 and EcoR1 each which cut lambda into 6 pieces of distinct lengths for a problem of size $6!6! = 518{,}000$, the algorithm was able to find the correct solution in 29,702, 6895, and 3670 iterations in runs from three different initial conditions. It is interesting to note that the solution to this actual problem was in fact unique. Further details may be found in [5].

References

[1] Bonomi, E. and J-L. Lutton (1984). The N-City travelling salesman problem: statistical mechanics and the Metropolis algorithm. *SIAM Review*, **26**, 551-568.

[2] Durand, R. and F. Bregerere (1985). An efficient program to construct restriction maps from experimental data with realistic error levels. *Nucleic Acids Res.*, **12**, 703-716.

[3] Fitch, W.M., T.F. Smith and W.W. Ralph (1983).Mapping the order of DNA restriction fragments. *Gene*, **22**, 19-29.

[4] Garey, M.R. and D.S. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completness. W.H. Freeman, San Francisco.

[5] Goldstein, L., and M. Waterman (1987). Mapping DNA by stochastic relaxation (1987). *Adv. Appl. Math.* **8**, 194-207

[6] Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6. 721-741.

[7] Hajek, B. (1985). Cooling schedules for optimal annealing. *Mathematics of Operations Research.*

[8] Kirkpatrick, S., C.D. Gelatt, Jr., M.P. Vecchi (1983). Optimization by simulated annealing. *Science*, **220**, 671-681.

[9] Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell Syst. Tech. J.*, **44**, 2245-2269.

[10] Metropolis, M., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087-1092.

[11] Nolan, C., G.P. Mairna and A.A. Szalay (1984). Plasmid mapping computer program. *Nucleic Acids Research*, **12**, 717-729.

[12] Pearson, W. (1982). Automatic construction of restriction site maps. *Nucleic Acids Res.*, **10**, 217-227.

[13] Stefik, M. (1978). Inferring DNA structure from segmentation data. *Artifical Intelligence*, **11**, 85-114.

[14] Wulkan, M. and T.J. Lott (1985). Computer aided construction of nucleic acid restriction maps using defined vectors. *Computer Appl. Biosc.*, **1**, 235-239.

VI. PARALLEL COMPUTING

Modeling Parallelism: An Interdisciplinary Approach

Elizabeth A. Unger
Department of Computing
and Information Sciences
Kansas State University

Sallie Keller-McNulty
Department of Statistics
Kansas State University

Abstract

One can easily conjecture that we humans have imposed sequential solutions onto most problems, such are a better match to our physical architecture, but we propose that there are parallel solutions to many problems and these are a better if they can be matched to our computer architectures. The discovery of problems involving parallelism in many and diverse disciplines which are the subject of current research efforts has been a simple matter, however the development of methods which discover the parallelism possible in solutions to a problem is not a simple matter and is the focus of this research. This paper will describe the model and discuss the current research efforts in terms of academic contributions and the strengths gained through the interdisciplinary group approach to problem solving.

At Kansas State University a group of people from three disciplines in two colleges has been formed to provide a critical mass of researchers and to create broader base of knowledge from which to draw to find an architecture-free model which can be used to express, in a natural way, the potential concurrency in problem solutions. A partially defined model based upon a conditioned dataflow which incorporates the concepts of control flow based on dataflow, of the description of an action at any level of detail with subsequent further refinement if desired, of repetition based upon partitions of data aggregates, of single assignment of values to uniquely identify each incarnation of data objects, and of partial computation, i.e., computation which can proceed until a needed unavailable datum is encounter has been developed. The group has four major foci to their work, 1) continuing development of the theoretical foundation of the model, led by the computer scientists, 2) use of the model to discover paradigm parallelism models for particular problems at the small and the large granularity levels of detail, led by the statistician and engineers 3) the development of methods of determining the best fit of the discovered parallelism to existing architectures, led by the statistician and engineers, 4) the continued implementation of a prototype on a distributed network of processors, led by the computer scientists. All members have contributed to all phases.

The current status of our work included a model which has been shown to contain a core of statements which always describe determinate problem solutions for atomic data types. A prototype is being used to study problem solutions where the granularity of the parallelism is small. On going research work involves providing the theoretical basis for temporally partitioned data aggregates, the inclusion in the prototype of partial computation, and limited data structures and the development of models of existing architectures using the model for the current multiprocessor architectures.

1. Introduction

Traditionally, computing machine design and the choice of problem solution has been predicated upon the sequential expression of computation. The advent of multiple processor architectures and computer networks requires a different approach to problem expression to fully utilize the available computational power. The primary component of this approach is the division of a problem solution into computational units and the ordering of the execution of these divisions. In this paper, a model/method will be developed which is based on the examination of the flow of data and on aggregation data to discover parallelism in numerical algorithms. This method of parallel computation seems to hold the great promise for statistical application (Lafaye de Micheaux, 1984).

Unlike much of the current research in parallel algorithms for statistical and numerical linear algebra problems, the model/method developed here is architecture independent [Heller 1978, O'Leary 1985,1980, Gokhale 1987]. Through the fundamental ideas of dataflow computation [Dennis 1972] and spatial and temporal partitioning of data structures into computationly independent units [Unger 1978] , the inherent parallelism in a problem solutions can be specified without the traditional concerns of communication, synchronization, data sharing and physical architecture [McBride 1983]. The architecture free approach to parallel computing is prompted by the idea that only after the inherent parallelism of a numerical method is expressed as grouping of the data (not necessarily limited to the data aggregates) does the architecture become a consideration. Jamieson (1987) calls this the virtual algorithm for a problem solution approach. Different architectures will give rise to different sequencings of the independent computational units from this virtual algorithm.

This paper is divided into three parts. Section 2 gives a description of the basic model/method we have developed. This section is followed by a discussion of the concepts of data and procedural abstraction. Section 4 deals with the notion of the existence of a virtual algorithm with a motivating example.

2. Basic Concurrency Model/Method

In this section a mathematically based model which can be used for concurrent computation is presented. This concurrency model/method is built upon the concept of representing both data and action as objects [Unger 1988b]. The fundamental principles of sequencing these objects will be illustrated as well as how this sequencing can be altered through the use of predicates.

Objects representing action can be aggregated into collections of objects resulting in more abstract action object or disaggregated into several less abstract action objects. Each action object can be represented as a 5-tuple (s,m,a,r,t) where s is a boolean predicate whose truth value determines when the action will be eligible for execution, m is a list of materials (input objects), a is the name or designator of the action, r is a list of results (output objects), and t is a boolean predicate whose truth value determines if and when the action should terminate prior to completion.

For illustrative purposes, the following syntactic form for an action object will be used,

$$[s]\ a(m;r)\ [t]\ .$$

where the elements of m and r are separated by commas. For example, computing the length of the hypothenuse of a right triangle using the Phythagorean Theorem could be expressed as shown in Figure 1.

Model Objects	Interpretation
Sqrt(temp3; c)	$\sqrt{a^2 + b^2} \rightarrow c$
Add(temp1, temp2; temp3)	$a^2 + b^2$
Sqr(a; temp1)	a^2
Sqr(b; temp2)	b^2

Figure 1: Hypothenuse Computation

The model is data driven. This means that the time at which an action is first eligible for execution is when all of the elements of m, the materials list, are available. Figure 2 gives the dataflow diagram [Petersen 1977, Karp 1966, McBride 1987, Noe 1973], for the hypothenuse computation of Figure 1. It is drawn such that an action appears at the first level, horizontally at which it is eligible for computation. At level 1 of Figure 2, there are two actions which can be computed concurrently, this represents the only inherent parallel computation in the example. Note that since the sequencing of computation is driven by the availability of the materials, or inputs, the order in which the syntactical statements are listed is immaterial. Thus allowing each problem solver freedom to conceive the problem solution in the most natural way for them.

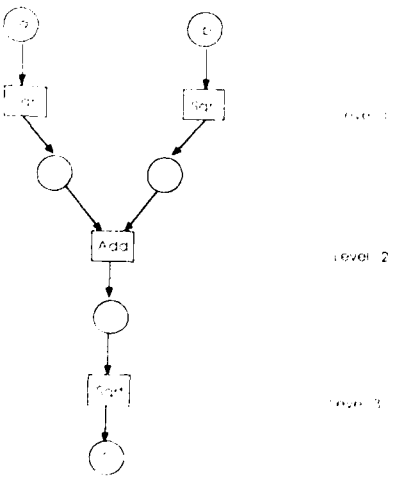


Figure 2: Data Flow for Hypothenuse Computation

Data objects in the model have two important components, the designator and the corporality type. The designator contains an arbitrary name assigned by the problem solver. Corporality or the length of existence of an object provides the capabilities to assure the determinacy of the problem solution results. Two corporality types of the model provide the concept of the single assignment of a value to an object [Comte 1976]. If the corporality type of a data object is static, only one value may ever be assigned to that data object. The default corporality of a data object is dynamic. When the corporality type of a data object is dynamic, the model adds a sequence indicator to the designator. This can be envisioned as the data object having a series of incarnations, each distinguished by the sequence indicator (e.g., x_i, x_{i+1}, x_{i+2}, ...) Objects with corporality type of dynamic can be referenced by their designator and sequence indicator or by their designator alone. If a dynamic object is referenced without a sequence indicator, the latest available incarnation of the object is retrieved. An additional corporality type of fluid is also defined by the model. A data object with the corporality type of fluid can change value with no incarnation indicator (like the common implementation of variable in current programming languages). Such objects are currently not allowed in the determinant subset of the model and will not be discussed further in this paper.

Figure 3 gives the calculation of a Fibannoci sequence of numbers as it would be described within the model. In this example the data object designated x has a dynamic type of corporality. The statements 1 and 2 indicate absolute references to the incarnations x_0 and x_1. The statement 3 references the data object x in a relative fashion directing that the previous incarnation is to be added to the current incarnation to form the next incarnation.

Model Objects	**Interpretation**	
declare x dynamic;		
assign (1; x.0)	$x_0 = 1$	[1]
assign (1; x.1)	$x_1 = 1$	[2]
add(x.-1, x; x.+1)	$x_{i+1} = x_i + x_{i-1}$	[3]

Figure 3: Use of Data Objects with Dynamic Corporality

The granularity of the action and data objects can vary. The smallest granularity action object are those that specify primitive actions, (e.g. +, -, *, /). Large granularity action objects are ones in which considerable detail must be provided in terms of the composing actions before primitive actions are specified. The action objects specified in Figure 1 are atomic actions, hence they have small granularity. If the action objects in Figure 1 were aggregated together into the action object Phythagorean(a, b; c), say, then this would be an example of an object with larger granularity. Syntactically this will be denoted as shown in Figure 4.

```
Phythagorean(a, b; c)
        { Sqrt(temp3; c)
        Add(temp1, temp2; temp3)
        Sqr(a; temp1)
        Sqr(b; temp2) } .
```

Figure 4: Phythagorean Action Object

This aggregate object has the designator, Phythogorean and two input (material) objects, a and b. The use of this aggregate object and the accompanying dataflow is shown is Figure 5.

Phythagorean ($side_1$, $side_2$; hypothenuse)

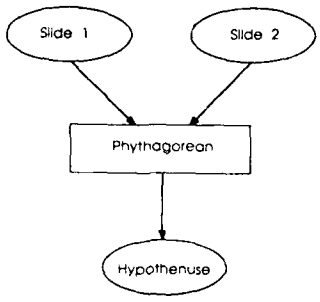


Figure 5: Aggregate action object Phythagorean

Values for each of the data objects, side1, and side2 are required for the action object Phythagorean to execute. We can use an aggregate data object say, A, composed of side1, and side2 and then the use of the action, Phythagorean could be expressed as shown below.

partition A: $side_1$, $side_2$
Phythagorean (A; hypothenuse)

The use of aggregate data objects allows us to form a partition of a data structure. For example, consider the pay check computation (CHECK) for a company with 100 employees and two computers. If the input to CHECK is pay_file which consists of 1000 records, we can aggregate the data into 2 aggregate data objects pay_1 and pay_2 as shown in Figure 6.

partition pay_1 : pay_file, records 1-500,
pay_2 : pay_file, records 501-1000.
the call for action would be :
CHECK (pay_1)
CHECK (pay_2)

Figure 6: Aggregate data object based on partitions

Predicates are used to govern when and if an action object is started or when an action object is terminated prior to completion of its specified task. Predicates which appear on a defined action object are termed internal conditions. Predicates used when an action object is requested (called into use) are termed external conditions. For example, an internal stimulation condition of [a=b] placed on the Phythegorean action object defined above limits the hypothenuse computation to isosceles triangles. Any attempted use of the Phythegorean object on non-isosceles triangles would result in no action. The action object Averages given below illustrates the use of an external simulation condition.

[count > 0] Averages(count, occurrences; $a_1 a_2 a_3$)
where the detail of Averages is: Averages (count, occurrences; $a_1 a_2 a_3$)
{ Mean(count, occurrences; a_1)
Median(count, occurrences; a_2)
Mode(count, occurrences; a_3) }

The action object Averages will be executed only if count (number of occurrences or data points) is greater than zero.

When executed, this action object returns three measures of central tendency, the mean, the median, and the mode. If one wished to merely have a measure of central tendency but did not care which one or did not want all three, internal termination conditions could be used on the object Averages as shown below.

[count > 0] Averages(count, occurrences; b)
{ Mean(count, occurrences; b) [b*]
Median(count, occurrences; b) [b*]
Mode(count, occurrences; b) [b*] }

The termination condition [b*] denotes the existence of a value for b. In this case the first action of mean, median , or mode to return a value for b will cause the other two actions to terminate immediately.

The model/method discussed in this section has a subset which will guarantee determinant behavior. Determinant behavior means that given same values for the input objects, the same values for the output objects will result. It should be noted that there are many situations, e.g., the above action object Averages, in which indeterminism is useful. A general insight into the determinant core is provided in Figure 7. If the model developed here is used on a computing system, deadlock potential exists. Also, the model requires there be exactly one viable source for each output this requirement may be difficult to ascertain.

No objects with longevity type fluid must exist.

No internal stimulation or external termination conditions may be used.

Number of requests in an action object must be finite.

At any level of abstraction, there can be only one viable source for each output object resulting from a request with an external stimulation condition.

Figure 7: General Conditions of the Determinant Core

3. Abstraction

A fundamental concept in this research is that inherent parallelism in a problem solution can be located by examining the problem solution at various levels of abstraction. This section explores the concepts of both action abstraction and data abstraction.

Benjamin Whorf (19) has said "Language shapes the thought and culture of those who use it." The model/method described in Section 2 provides an environment or language that encourages abstraction by its syntactic constructs and structure. Top-down statement of solutions to problems is encouraged through the concept of detailing or disaggregation of objects. Bottom-up statement of solutions to problems is encouraged through the concept of aggregation or construction of objects.

Detailing of an object involves the replacement of the object with a set of smaller granularity objects expressing the same action or data, only in more detail. Detailing can continue in a problem solution until either an interface with an existing object occurs or an interface with a computational device occurs. Aggregation or construction is the reverse of detailing. Aggregation is the process of defining a structure or collection of one or more objects. The basic operations on the collection are defined within the aggregate and are the operations used when instantiations of the collection are manipulated.

Parallelism in problem solutions can be discovered and described by examining the way in which aggregates or collections of data objects can be manipulated. An aggregation of data which results in aggregate tokens (or groups) of the data objects which are computationally independent represents a set of data aggregates which can be scheduled in parallel or concurrently. These aggregate tokens can be homogeneous, like in type and semantic meaning, or nonhomogeneous. We will restrict our discussion here to homogeneous aggregate tokens.

A series of examples from both office automation and numerical linear algebra will be used to demonstrate the concept of abstraction in a problem solution. It is interesting to note that the model/method developed in Section 2 is equally effective in both of these areas.

The payroll check calculation example (See Figure 6) is one example of a transaction processing problem solution. Transaction processing means that the calculation of each unit of computation, e.g., a payroll check, is independent of the computation of all others units. In such situations the input can be divided by partitioning the input file in any fashion without affecting the output results. There are consequences of the level of aggregation. For instance if the pay-file was divided into 1000 aggregates which form a partition then one could cause 1000 different aggregates to be sent to other processors. While in Figure 6 there are only 2 data object aggregates (tokens) which can be sent, thereby reducing communication overhead.

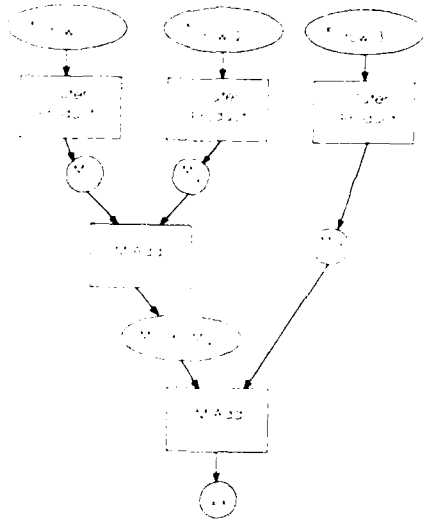
There is no need for the data aggregates to form a partition of the data although the potential for indeterminate computation may occur if the problem solver creates code outside of the determinant core. For instance, there could be more than one source for a given result (see the Averages example).

Consider examples, this time involving a two dimensional collection of homogeneous data objects which form a matrix. In parallel solutions to numerical linear algebra problems, the questions of what computation can be done in parallel and what degree of aggregation of the data should be used arise. The first question deals with locating the inherent parallelism in a problem solution. The second question addresses the issue that a particular numerical method will be of interest to people dealing with both small and large dimensional matrices and the fact that one cannot expect to have an infinite number of processors available.

The repetitive computation of initializing each element of the matrix to the same value can be thought of as creating aggregate tokens which contain exactly one element of the matrix and scheduling the entire initialization to occur concurrently. This solution represents the maximum amount of potential parallelism. In many situations, the dimension of the matrix will greatly exceed the number of available processors or this small level of granularity will be impractical because of interprocessor communication costs. Another solution to this problem would be to aggregate pieces of the matrix into aggregate tokens and to perform the initialization on the tokens in the aggregate tokens sequentially. The initialization of each aggregate token can occur in parallel.

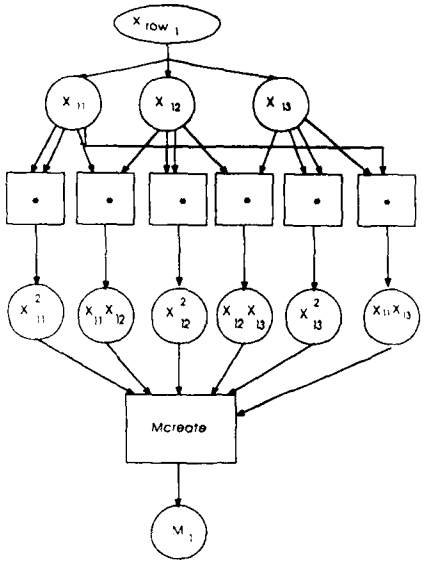
A variation on the previous example would be the initialization of a matrix to the identity matrix. Again the maximum degree of parallelism would occur by letting each element form an aggregate token and initialize everything at once assuring that the partitions containing the diagonal elements were assigned a one and everything else a zero. An alternate aggregation of the matrix elements could consist of forming an aggregate token that contains the diagonal elements and one or more aggregate tokens that collects together the off-diagonal elements of the matrix. Initialization would then occur sequentially within each aggregate token.

The calculation of $X^t X$ using this outer product is an example where one can consider the solution at several levels of aggregation; for simplicity we illustrate this with X, a 3 X 3 matrix. Figure 8 illustrates the calculation based upon data object aggregates which are rows. Figure 9 is a detail of the outer product calculation for the first row of the matrix X. Clearly if X were larger we might group the rows together as shown in Figure 10 and then send these aggregate token to different processors for potentially concurrent computation.



Madd - is an action that is an element by element add for matrices

Figure 8: Outer product for $X^t X$ based or row partitions



where M_1 is

$$\begin{bmatrix} X_{11}^2 & X_{11}\ X_{12} & X_{11}X_{13} \\ X_{11}\ X_{12} & X_{12}^2 & X_{12}X_{13} \\ X_{11}X_{13} & X_{12}\ X_{13} & X_{13}^2 \end{bmatrix}$$

Figure 9: Detail of outer product for first row of X

$$\left.\begin{matrix} X_{11} \cdots & X_{1n} \\ X_{21} \cdots & X_{2n} \\ X_{31} \cdots & X_{3n} \end{matrix}\right\} XR_{13}$$

$$\left.\begin{matrix} X_{41} \cdots & X_{4n} \\ X_{51} \cdots & X_{5n} \\ X_{61} \cdots & X_{6n} \end{matrix}\right\} XR_{46}$$

Figure 10: Data aggregate of 3 rows

4. Virtual Algorithm

Jamieson (1987) proposes that for any problem solution approach there is a virtual algorithm. She also proposes that this virtual algorithm can be mapped to one of a number of architecture specific algorithms. Jamieson's Virtual Algorithm Approach is depicted in Figure 11.

The virtual algorithm, for those problem solutions that require no iterative computation, is defined by mapping the inputs directly to the outputs, recognizing the renaming and use of the inputs in intermediate computations. In terms of the methodology discussed in this paper, this means expressing the problem solution with the finest degree of detail and complete disaggregation of the data objects. Obviously this is a formidable task and two practical questions arise. First, is finding the virtual algorithm useful? Second, since detailing is the reverse of aggregation, is it possible to gleen the useful information from the virtual algorithm expressed at a higher level of abstraction?

If virtual algorithms could be found and expressed in a reasonable way, all the inherent parallelism in a numerical method could be understood and all possible sequencings of the computations could be defined. We will use a graphical representation of the Cholesky decomposition of a matrix to study the usefulness of a virtual algorithm. The answer to the second question remains open.

First we will consider the dataflow of two well known Cholesky decomposition algorithms. The first is a traditional method given in Figure 12a and b and the second given in Figure 13a and b was discussed by O'Leary and Stewart (1986). Neither of these dataflow diagrams represent the virtual algorithm for this numerical method in that the renaming and use of the original inputs has been ignored. Figure 14 represents the virtual algorithm for this numerical method. Within the dataflow graph of Figure 14, each primitive action on the original inputs is represented at the earliest time frame (level) in which the input for that action is available and for which the corresponding predicates are satisfied. The diagram of the virtual algorithm maintains the vertical positioning of actions corresponding to time. Observe that the dataflow of the traditional algorithm of Figure 11b is equivalent to sequencing the computation according to horizontal planes cutting the diagram at each time frame. The O'Leary and Stewart method of Figure 13b is also evident in the virtual algorithm diagram. That computation proceeds in the order given by the vertical planes shown in Figure 14.

5. Conclusions

This research is directed toward the discovery of inherent parallelism (or the virtual algorithm) for a given problem solution approach. A concurrent method/model which has a graphical form and a linear syntactic form has been presented which can be used as a tool for parallel algorithm development. One advantage of the location of such virtual algorithms is the potential of mapping these algorithms to optimal architecture specific algorithms.

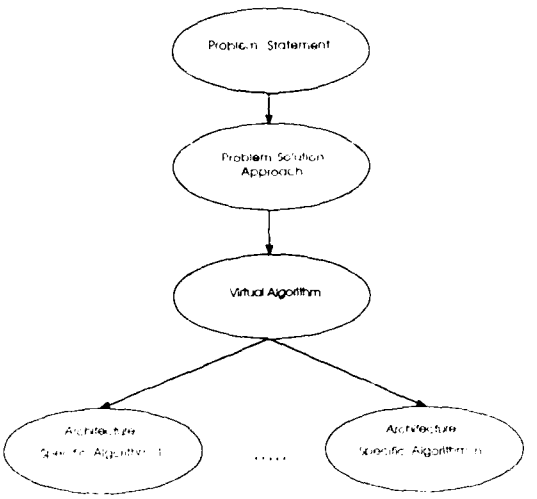


Figure 11: Virtual Algorithm Approach

for k = 1,2, . . . , n

$$l_{kk} = \left(a_{kk} - \sum_{p=1}^{k-1} l_{kp}^{2} \right)^{1/2}$$

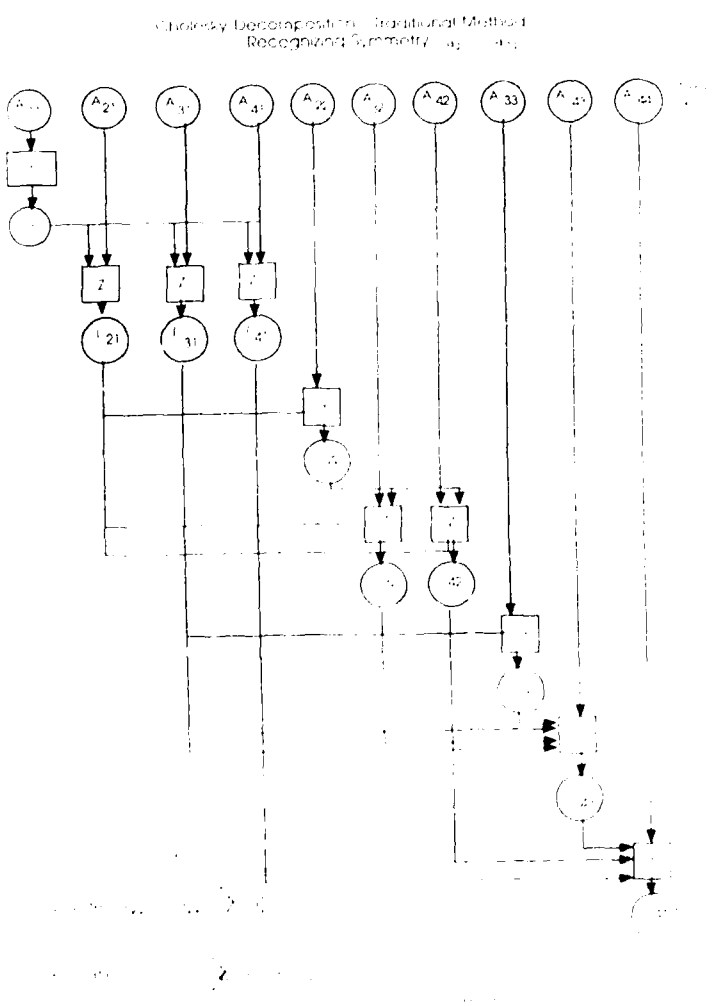
for i = k + 1, . . . , n

$$l_{ik} = \left(a_{ik} - \sum_{p=1}^{k-1} l_{ip} l_{kp} \right) / l_{kk}$$

end for

end for

Figure 12a: Traditional Algorthim for Cholesky Decomposition



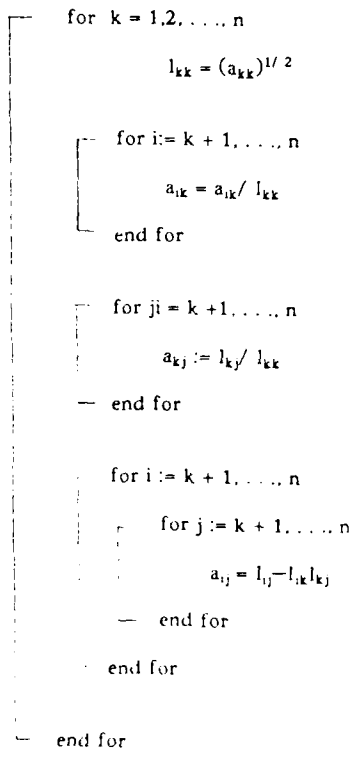


Figure 13a: Stewart and O'Learys' (1985) algorithm for Cholesky Decomposition

Cholesky Decomposition - O'Leary & Stewart (1985)

Assumes symmetry ($a_{ij}=a_{ji}$)

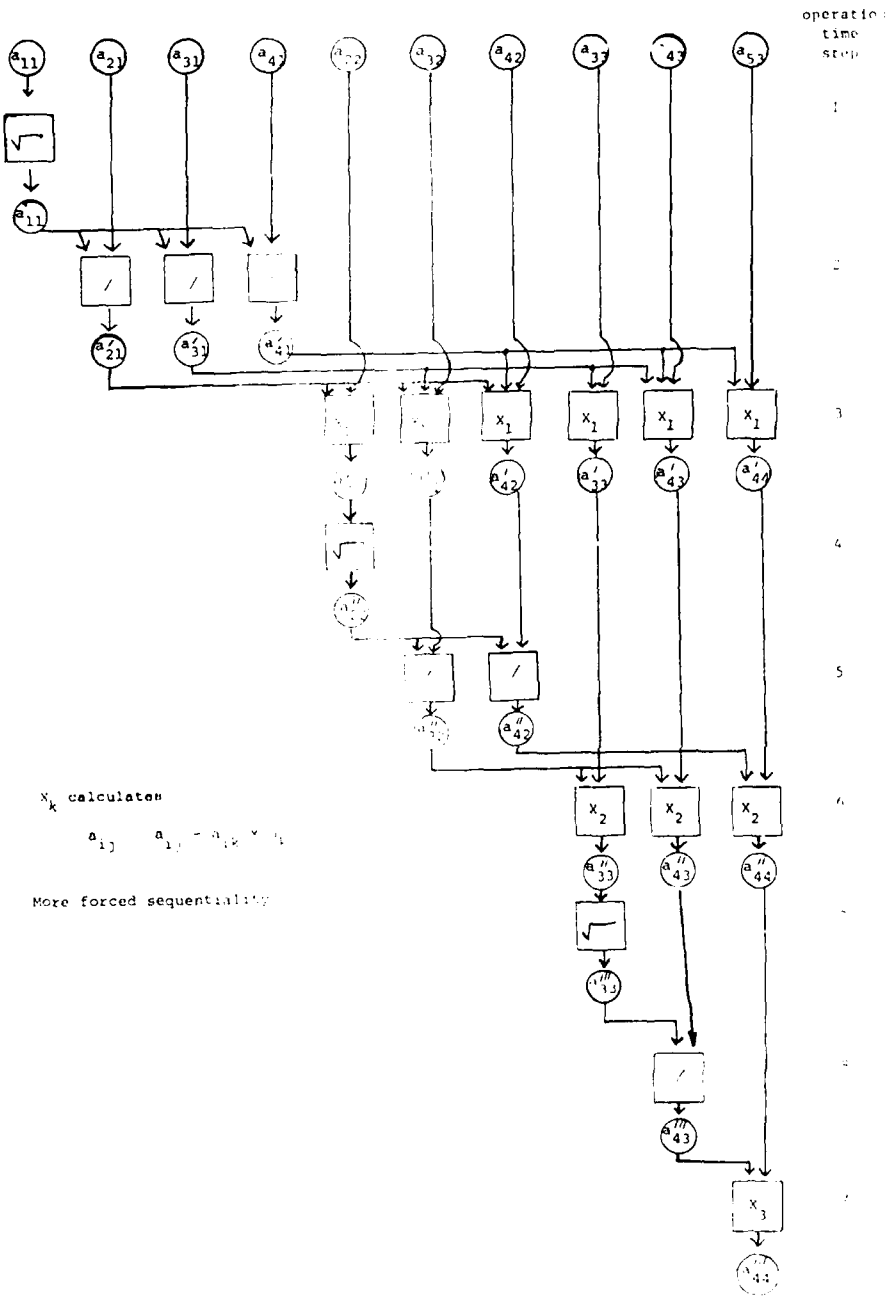


Figure 13b

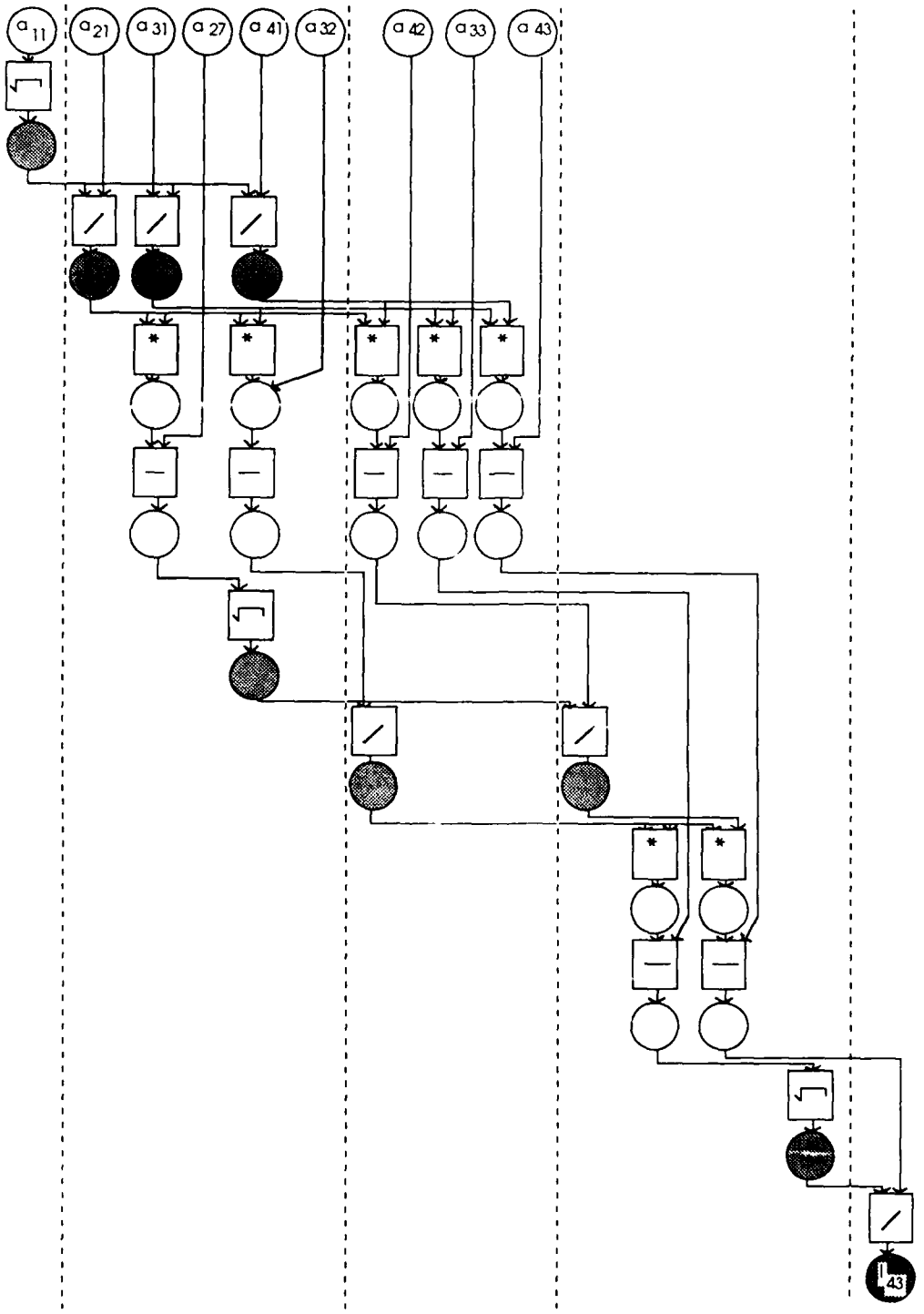


Figure 14 Cholesky Decomposition - Virtual Algorithm for l_{43}

Bibliography

Comte, D. et. al. (1976), "Parallelism, Control, and Synchronization Expression Single Assignment Language," Proceedings Fourth Annual ACM Computer Science Conference

Dennis, J.B., Fosseen, J.P. and Lindman, J.P. (1972), "Data Flow Schemes," Project MAC Computational Structures Group Memo, MIT Press, Cambridge, Mass..

Gokhale, M.B. (1987), "Exploiting Loop Level Parallelism in Nonprocedural Data Flow Programs," NASA Contractor Report 178277, ICASE Report No. 87-23.

Jamieson, L.H., (1987), "Characterizing Parallel Algorithms," The Characteristics of Parallel Algorithms, MIT Press, Cambridge, Mass., 65-100.

Heller, D.E. (1978), "A Survey of Parallel Algorithms in Numerical Linear Algebra," SIAM Review, 20, 740-777.

Karp, R.M., and Miller, R.E. (1966), "Properties of a Model for Parallel Computations: Determinancy, Termination, Queueing," SIAM Journal on Applied Mathematics, 14, 1390-1411.

—— (1969), "Parallel Program Schemata," Journal of Computer and System Sciences, 3, 147-195.

Lafaye de Micheaux, D. (1984), "Parallelization of Algorithms in the Practice of Statistical Data," COMPSTAT 1984, International Association for Statistical Computing, pp. 293-300.

McBride, R.A., and Unger, E.A. (1983), "Modeling Jobs in a Distributed System," ACM SIGPC Notes, 6, 32-42.

—— (1987), "The Representation and Distribution of Knowledge by a Petri Net," in Proceedings of the Fifth International Conference on Systems Engineering, IEEE.

Noe, J.D. and Nutt, G. (1973), "Macro E-Nets for Representation of Parallel Systems," IEEE Transactions on Computers, C-22, 718-727.

O'Leary, D.P. and Stewart, G.W. (1986), "Assignment and Scheduling in Parallel Matrix Factorization," Linear Algebra and Its Applications, 77, 275-299.

—— (1985), "Data-FLow Algorithms for Parallel Matrix Computations," Communications of the ACM, 28, 840-853.

Peterson, J.L. (1977), "Petri Nets," Computing Surveys, 9, 223-252.

Unger, E.A. (1978), "A Concurrent Model: Basic Concepts," Data-Flow Workshop Proceedings, European Conference in Parallel and Distributed Processing.

Unger, E.A., Honeyman, J., Van Swaay, M. and Va, R. (1988), "Intelligent Data Object: A Model for the Control and Manipulation of Data Objects in a Network Environment," Proceedings of Second Oklahoma Conference on Applied Computing.

Unger, E.A., Hsieh, S. and Van Swaay, M. (1988), "A Concurrency Method Prototype Implementation," Kansas State University Computer Science Technical Report.

ASYNCHRONOUS ITERATION

William F. Eddy *Mark J. Schervish*
Department of Statistics, Carnegie-Mellon University

1 Introduction and Summary

Solutions to fixed point problems, solutions of equations, and maximizations often involve iterative schemes. When each iteration consists of evaluating a vector of functions, the possibility exists for evaluating the coordinates of that vector *asynchronously*, that is, not necessarily all at the same time. For example, consider the following iterative method of finding the largest eigenvalue and corresponding eigenvector for a symmetric $m \times m$ matrix A, starting with a vector $\mathbf{x}^0$:

$$\begin{aligned} \mathbf{y} &= A\mathbf{x}^n = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \\ c_{n+1} &= \max_j y_j \\ \mathbf{x}^{n+1} &= c_{n+1}^{-1}\mathbf{y}. \end{aligned}$$

Under certain general conditions, the sequence $c_1, c_2, \ldots$ is known to converge to the eigenvalue of A with largest absolute value and the sequence $\mathbf{x}^1, \mathbf{x}^2, \ldots$ converges to a corresponding eigenvector. Now suppose that m is even and we write

$$A = \begin{pmatrix} A_0 \\ A_1 \end{pmatrix},$$

where each of A_0 and A_1 are $(m/2) \times m$ matrices. We can form the sequence of "partial" iterations $\mathbf{x}^1, \mathbf{x}^2, \ldots$, where

$$\begin{aligned} \mathbf{x}^n &= \begin{pmatrix} \mathbf{x}_0^n \\ \mathbf{x}_1^n \end{pmatrix} \\ \mathbf{y} &= A_{(n \bmod 2)}\mathbf{x}^n = \begin{pmatrix} y_1 \\ \vdots \\ y_{\frac{m}{2}} \end{pmatrix} \\ c_{n+1} &= \max_j y_j \\ \mathbf{x}^{n+1} &= \begin{cases} \begin{pmatrix} c_{n+1}^{-1}\mathbf{y} \\ \mathbf{x}_1^n \end{pmatrix} & \text{if } n \text{ is even,} \\ \begin{pmatrix} \mathbf{x}_0^n \\ c_{n+1}^{-1}\mathbf{y} \end{pmatrix} & \text{if } n \text{ is odd} \end{cases} \end{aligned}$$

Each iteration here only calculates new values for half of the vector, keeping the other half the same as the previous iteration.

An obvious question arises as to what is to be gained by such a partial iteration scheme. For example, in the eigenvalue calculation, might it be that it takes fewer than twice as many "half iterations" to achieve the same degree of convergence? Could it be that the partial iterations do not even converge? In this paper, we provide some theoretical and some empirical answers to questions of this sort. One situation in which partial iterations have a great deal of potential is in a parallel/distributed computing environment. For example, in the eigenvalue calculation, if one had two processors available, one could assign all iterations invoving A_0 to one processor and all iterations involving A_1 to the other one. The sequence of iterations would not be the same as that described above if both processors were allowed to work at the same time. The reason is that iterations n and $n+1$ might be proceding simultaneously, hence iteration $n+1$ could not be a function of iteration n. If the processors ran at different speeds, the iterations would not even alternate between the two halves of the vector. For this reason, such a sequence of iterations is *asynchronous*.

In Section 2, we give a precise definition of asynchronous iterations and the types of problems in which they have been applied. In Section 3, we give examples of some asynchronous iteration schemes which can be used in most iterative problems. In Section 4, we present some theorems giving conditions under which asynchronous iterations converge. In Section 5, we desribe the example calculations we performed. These calculations are all based on the eigenvalue problem described above. In Section 6, we briefly describe a video animation system and some videotapes we created to help visualize the sequence of asynchronous iterations.

2 Definitions and Notation

Consider a mapping from a subset D of n-dimensional Euclidean space $\Re^n$ to $\Re^n$,

$$F = (F_1, \ldots, F_n) : D \to \Re^n$$

We will consider the problem of finding a fixed point of this mapping by means of successive iteration. The idea is that, since a fixed point $\mathbf{x}$ satisfies $F(\mathbf{x}) =$

$\mathbf{x}$, then starting at an arbitrary point $\mathbf{x}^0$, we could successively calculate $\mathbf{x}^j = F(\mathbf{x}^{j-1})$. If the sequence $\{\mathbf{x}^j\}_{j=0}^{\infty}$ converges, it must converge to a fixed point.

Here, we consider a more general sequence of iterations called *asynchronous iterations*.

Definition 1
A sequence of iterations $\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^j, \ldots$ are called *asynchronous iterations* if

$$x_i^j = \begin{cases} x_i^{j-1} & i \notin L_j \\ F_i(x_1^{s_1^j}, \ldots, x_n^{s_n^j}) & i \in L_j, \end{cases}$$

where L_j is a nonempty subset of $\{1, \ldots, n\}$ listing the components of $\mathbf{x}$ updated at iteration j and the numbers s_i^j are integers indicating which iterate of x_i to use at iteration j. We require that $s_i^j \leq j-1$ so that the procedures can be realized in practice.

For convenience we define

$$\begin{aligned} \mathbf{L} &= \{L_j | j = 1, 2, \ldots\} \quad \text{and} \\ \mathbf{S} &= \{(s_1^j, \ldots, s_n^j) | j = 1, 2, \ldots\}, \end{aligned}$$

The types of functions F we will consider are *Lipschitzian contractions*. These are a special class of *Lipschitzian operators*.

Definition 2 *A function $F : D \rightarrow \Re^n$ is a Lipschitzian operator* if there exists an $n \times n$ matrix A with non-negative entries such that $|F(\mathbf{x}) - F(\mathbf{y})| \leq A|\mathbf{x} - \mathbf{y}|$ where $|\cdot|$ and $\leq$ are taken component-wise. The matrix A is called the *Lipschitzian matrix* for the operator F.

Definition 3 A function $F : D \rightarrow \Re^n$ is a *Lipschitzian contraction* if F is a Lipschitzian operator with matrix A having $\rho(A) < 1$, where $\rho(\cdot)$ is the spectral radius of its matrix argument.

3 Examples

In this section, we present examples of this general class of iterations, some of which can take advantage of distributed computation. Because the definition of Lipschitzian contraction requires the operator to be essentially linear (at least locally) we will consider here iterations of the form

$$x_i^j = \begin{cases} x_i^{j-1} & i \notin L_j \\ A_i \begin{pmatrix} x_1^{s_1^j} \\ \vdots \\ x_n^{s_n^j} \end{pmatrix} & i \in L_j, \end{cases}$$

where A_i is the i^{th} row of a matrix A.

3.1 Jacobi Iterations

Jacobi iteration is a standard procedure for solving linear systems and is sometimes called the method of simultaneous displacements. We describe a few of its variants here.

Sequential vector-wise evaluation: This iterative scheme is designed to run in a single process. We use

$$\begin{aligned} L_j &= \{1, \ldots, n\} \\ s_i^j &= j - 1. \end{aligned}$$

Here, every coordinate of F is evaluated at the same vector $\mathbf{x}^{j-1} = (x_1^{j-1}, \ldots, x_n^{j-1})$ and then the entire vector is updated to $\mathbf{x}^j = (x_1^j, \ldots, x_n^j)$. Thus, n components are updated per iteration. This is the standard method of iteration mentioned at the beginning of Section 1.

Independent component-wise evaluation: If n processes are available, each one can be devoted to updating a separate coordinate. That is, we use

$$\begin{aligned} L_j &= \{(j-1 \bmod n) + 1\} \\ s_i^j &= \lfloor \frac{j-1}{n} \rfloor n. \end{aligned}$$

What happens here is that no updated coordinate is used in any iteration until every coordinate has been updated. That is, every iteration consists of updating a single coordinate, and then every n iterations all of the updated coordinates become available for future iterations. Thus, one component is updated per iteration and the processes must be "synchronized" after every n iterations; that is, the $n+1$st iteration cannot proceed until the first n have all been completed.

Independent block-wise evaluation: If k processes are available and $n = qk$, then each process can evaluate q coordinates at a time. Here we use

$$\begin{aligned} L_j &= \{i | mq + 1 \leq i \leq (m+1)q\} \\ s_i^j &= \lfloor \frac{j-1}{n} \rfloor q \end{aligned}$$

where $m = (j - 1 \bmod k)$. Instead of updating only one coordinate per iteration, we evaluate q coordinates per iteration. But we still wait until all n coordinates get updated the same number of times before releasing the updated values for use by the next iteration. Thus, q components are updated per iteration and the processes must be synchronized after every k iterations.

3.2 Gauss-Seidel Iterations

Gauss-Siedel iteration is another standard procedure for solving linear systems and is sometimes

called the method of succesive displacments. Gauss-Siedel iteration is generally considered preferable to Jacobi iteration for solving linear systems. We describe a few of its variants here.

Sequential component-wise evaluation: At each iteration a single component of F is updated making use of the most recent values of all the other components. In this method, we do not wait for every n^{th} iteration to release updated coordinates for use by the next iteration. We use

$$\begin{aligned} L_j &= \{(j-1 \bmod n)+1\} \\ s_i^j &= j-1. \end{aligned}$$

That is, the coordinates are updated in sequence, one at a time, but as soon as a coordinate is updated, it is used in all future iterations. In the Jacobi methods, it was always the case that, for j sufficiently large, $\mathbf{x}^j = F(\mathbf{x}^l)$ for some $l < j$. In general, this will not be true for Gauss-Seidel iteration.

Sequential block-wise evaluation: At each iteration a block of coordinates of size q is updated making use of the most recent values of all the components not in the block. At the end of each iteration the new coordinates are leased for use in calculating the iterates for other blocks. We can use

$$\begin{aligned} L_j &= \{i | mq+1 \le i \le (m+1)q\} \\ s_i^j &= j-1 \end{aligned}$$

where $m = (j-1 \bmod k)$. At iteration j each coordinate in a block is updated based on the same starting vector $\mathbf{x}^{j-1}$. All future iterations make use of these updated coordinates and q components are updated at each iteration.

3.3 Random Iterations

Randomness can enter into iterative schemes in one or both of two ways. Either the components to be updated, L_j, may be uncertain, or the iterates to use $\{s_i^j\}$ may be uncertain, or both. The reasons why either or both of these items is uncertain may vary from one iterative scheme to the next. We will describe two such schemes. In each of the schemes described below, the randomness enters solely through uncertainty about the order of completion of the iterations. For this reason, we introduce the set C_j as the set of indices of all iterations which have completed at the time that iteration j begins. For example, in the sequential evaluation schemes described in Section 3.1 and Section 3.2, C_j is always $\{1, \ldots, j-1\}$. In the independent (Jacobi) schemes, C_j would be a proper subset of $\{1, \ldots, j-1\}$. For brevity, we only describe block-wise evaluation schemes in this section because component-wise schemes are special cases with one component per block.

Asynchronous fixed block-wise evaluation: If k processes are available, separate the $\{1, \ldots, n\}$ into k disjoint blocks. Each process can be assigned one of the blocks of coordinates. Each process updates the same coordinates at each of its iterations, using the latest available iterates of all coordinates. A process begins a new iteration as soon as it finishes an old one. When each block has q coordinates, so that $n = qk$, we can express this by

$$\begin{aligned} L_j &= \{i | (j-1)q+1 \le i \le jq\},\ 1 \le j \le k \\ L_j &= L_J,\ j > k \\ s_i^j &= \max\{k \in C_j | i \in L_k\}, \end{aligned}$$

where J is the random iteration number of the most recently completed iteration. Here, the block of coordinates to be updated at iteration j is uncertain due to the fact that we do not know which of the k ongoing iterations (processes) will finish next (and hence begin the next iteration). After each iteration completes, the newly updated coordintates become available for use at all future iterations.

Asynchronous cyclic block-wise evaluation: If k processes are available, and $n = kq$, the coordinates are divided into blocks of size q and the blocks are updated cyclicly. Each iteration consists of updating the next block of coordinates. Each process uses the latest available iterates of all coordinates. A process begins the next iteration in sequence as soon as it finishes an old one. We can express this by

$$\begin{aligned} L_j &= \{i | mq+1 \le i \le (m+1)q\} \\ s_i^j &= \max\{k \in C_j | i \in L_k\} \end{aligned}$$

where $m = (j-1 \bmod k)$. Here, the block of coordinates to be updated at iteration j is known due to the cyclic nature of the scheme. But which iterate of each coordinate to be used in the next iteration is uncertain until we know which previous iterations have finished. After each iteration completes, the newly updated coordintates become available for use at all future iterations.

Obviously, none of the block evaluation schemes require that $n = kq$. However, L_j is simpler to express when $n = kq$.

4 Theoretical Results

Several authors have proven that asynchronous iterations converge under certain conditions. These conditions generally involve the number of times each coordinate is updated, and how large s_i^j gets. In

Section 4.1, we present two previous results on the convergence of asynchronous iterations which impose deterministic criteria on the performance of the iteration scheme. In Section 4.2, we discuss probabilistic criteria which lead to almost sure convergence of asynchronous iterations.

4.1 Deterministic Results

The following conditions are assumed in the first two theorems.

1. $\lim_{j\to\infty} s_i^j = \infty, \forall i$
2. $i \in L_j$ infinitely often.

The first of these conditions guarantees that the ultimate step of the iteration depends on penultimate steps rather than very old steps. The second of these conditions guarantees that every component will be updated many times. Chazan and Miranker (1969) proved a theorem concerning affine functions.

Theorem 4 Chazan and Miranker (1969). *If $F(\mathbf{x}) = A\mathbf{x} + b$ then the asynchronous iteration converges if and only if $\rho(A) < 1$.*

Baudet (1975) was concerned with Lipschitzian contractions.

Theorem 5 Baudet (1975). *If F is a Lipschitzian contraction then the asynchronous iteration converges to the unique fixed point of F.*

The third theorem, due to Lubachevsky and Mitra (1986), applies only to finding the fixed point of a matrix $A = ((a_{i,i}))$ with $\rho(A) = 1$. Here $F(\mathbf{x}) = A\mathbf{x}$. In this theorem, for each $i \in L_j$, s_k^j is allowed to depend on i. That is,

$$x_i^j = \begin{cases} x_i^{j-1} & i \notin L_j \\ F_i(x_1^{s_1^j(i)}, \ldots, x_n^{s_n^j(i)}) & i \in L_j. \end{cases}$$

Theorem 6 Lubachevsky and Mitra (1986). *Suppose A is a non-negative irreducible matrix and assume there is i such that $a_{ii} > 0$, $x_i^0 > 0$, and $s_i^j(i) = j - 1$ for all $j > 0$. Then the asynchronous iteration converges to a scalar multiple of the fixed point of A.*

4.2 Probabilistic Results

There are two types of probabilistic results with which we will deal. The distinction depends on the relationship between the way L_j is chosen and the times taken to complete the first $j-1$ iterations. We define the j^{th} *service time* to be the time from the start of the j^{th} iteration until its completion. The first type of result deals with the case in which the L_j are chosen independently of the service times. The second type of results allow the L_j to depend on the service times. Throughout this section, we assume that the service times are finite almost surely. We will also assume that $s_i^j = \max\{k \in C_j | i \in L_k\}$, so that there is no chance of a coordinate "getting stuck" at an old value when newer updates are available. The only thing required in order to guarantee that the two conditions at the beginning Section 4.1 will hold with probability 1 is that

$$\begin{aligned} &\Pr(i \in L_j, \text{ for infinitely many } j) \\ &\quad = \quad 1 \text{ for each } i = 1, \ldots, n. \end{aligned} \tag{1}$$

We will consider schemes which guarantee (1) both when L_j is independent of the service times and when L_j depends on the service times.

4.2.1 L_j Independent of Service Times

Here we will describe some schemes which are designed to guarantee (1). The basic idea of these schemes is to choose the L_j $j = 1, \ldots$ in such a way that each coordinate has a positive probability of being in L_j for $j = k, \ldots, k+m$ for all sufficiently large k and some finite m, and to be sure that the probability of each coordinate being in L_j does not go to 0 as j increases. In this case, the law of large numbers will assure that each i appears in infinetly many L_j with probabilty 1. One way to arrange this would be to choose a collection of r subsets of $\{1, \ldots, n\}$, say $M_1, \ldots, M_r$, such that

$$\{1, \ldots, n\} = \bigcup_{t=1}^{r} M_t.$$

Then let L_j be a random choice from $M_1, \ldots, M_r$. If the choices are made independently and

$$p_t = \Pr(L_j = M_t) > 0$$

for each t and all j, then the law of large numbers guarantees that (1) holds with probability 1.

There is another class of schemes, which we will call *Markov schemes*, which also guarantee (1). If we let

$$p_{t,s} = \Pr(L_j = M_t | L_{j-1} = M_s)$$

for all $j > k$, then we can state some sufficient conditions for (1) to hold. For example, if the transition matrix $P = ((p_{t,s}))$ is regular (i.e. P^m has all nonzero entries for some m) then (1) holds because each coordinate has some positive probability of appearing in at least one of the next m L_j, and the probability

does not go to 0 as j increases. Also, if $P \neq I$, but $P^m = I$ for some m, then (1) holds. Asynchronous cyclic block-wise evaluation is such as scheme. It corresponds to $r = k$ and

$$p_{t,s} = \begin{cases} 1 & \text{if } t = (s+1 \bmod k) \\ 0 & \text{otherwise.} \end{cases}$$

In this case $P^k = I$. Many other block-wise schemes are available among the Markov schemes, including both deterministic and random choices of L_j.

4.2.2 L_j dependent on Service Times

When the L_j are dependent on the service times, various difficulties can arise. For example, a silly algorithm for choosing L_j would be, for $j > 10$, if any of the first 10 completed service times is greater than 14 seconds, $L_j = \{1\}$. Assuming that the service time distribution had postive probability beyond 14 seconds, there would be positive probability that all coordinates other than 1 would be updated only finitely often. Rather than try to construct necessary conditions for ruling out this type of behavior, we propose simple sufficient conditions.

Suppose that we choose r subsets of $\{1, \ldots, n\}$, say $M_1, \ldots, M_r$, such that

$$\{1, \ldots, n\} = \bigcup_{t=1}^{r} M_t,$$

and each L_j is required to be one of the M_t. One way in which the L_j can be dependent upon the service times is for L_j to be a function of which M_t was updated in the iteration which most recently completed.

Asynchronous fixed block-wise evaluation is an example of this type of scheme, in which L_j is exactly that M_t which was updated by the iteration which most recently completed. This requires that $r = k$ and that the first k of the L_j are $M_1, \ldots, M_k$ in some order. This scheme has the property that (1) holds. There are other such schemes for which (1) holds. That is, suppose that $r = k$ and that the first k of the L_j are $M_1, \ldots, M_k$ in some order. Let

$$f : \{1, \ldots, k\} \rightarrow \{1, \ldots, k\}$$

be a one-to-one function. For $j > k$, let m_j be the number of the iteration which completes just before iteration j begins. Then (1) will hold if $L_j = M_{f(p)}$, where $L_{m_j} = M_p$. There are $k!$ such schemes and asynchronous fixed block-wise evaluation corresponds to $f(i) = i$, for $i = 1, \ldots, k$.

5 Empirical Results

In this section we describe the test cases which we ran to compare the performance of synchronous and asynchronous iterations on a parallel/distributed system. The particular system of processors used in the computations is described by Eddy and Schervish (1986) and has been used in several statistical applications (Eddy and Schervish, 1987 and Schervish, 1988). A brief description follows.

5.1 The Distributed System Used

The parallel/distributed system used in the examples of this paper is a special case of a *master-slave* system. In a master-slave system, one process acts like a master, keeping track of control information, such as **L** and **S** and which iterations are outstanding. The slave processes perform the bulk of the numerical calculations, such as function evaluations and matrix multiplications. The system of Eddy and Schervish (1986) uses the DECnet communication protocol between VAX computers running the VMS operating system. The master process communicates with the slaves by writing to and reading from *network devices* (DECnet's way of defining communication channels).

Data-flow is implemented by having some of the reading and writing done *asynchronously*. For example, the master begins by assigning a task to each slave. This is done by writing the appropriate data and/or instructions to the network device associated with each slave. The master then reads from the network device, but does not wait for a response. Figuratively speaking, the master says "Let me know as soon as something arrives." Then the master goes on to the next slave. When "something arrives" from a slave, the master deals with the response and sends another task (if any remain) in the same way as before. On the other hand, each slave begins by reading from the network device and waiting for a task to arrive from the master. It then does its work, writes its response to the network device, and waits for another task. When the work is finished, the master can release the slaves or keep them waiting for a brand new set of tasks.

5.2 The Example Matrix

We used three different iterative schemes for finding the largest eigenvalue and corresponding eigenvector (henceforth called the *largest eigenvector*) of a matrix A. The matrix is a 499×499 circulant with $(.9)^{|i-j|}$ in the (i, j) entry. The iterative methods we used were based on the iterative algorithm described

in Section 1 and letting $\mathbf{x}^0$ be any vector not orthogonal to the largest eigenvector.

The matrix used in the example has a simple eigenstructure which we describe here. These results follow from the theorems of Section 6.5.2 of Anderson (1971). Our matrix A can be expressed as

$$A = I + \sum_{i=1}^{249} (.9)^i A_i$$

where A_i is a matrix whose only non-zero entries are 1s on the i^{th} and $(250-i)^{\text{th}}$ sub- and super-diagonals. That is, if $A_i = (a_{jk}^{(i)})$, then

$$a_{jk}^{(i)} = \begin{cases} 1 & \text{if } |j-k| = i \\ 1 & \text{if } |j-k| = 250 - i \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 6.5.3 of Anderson (1971) says that the eigenvalues of A_i are

$$\begin{aligned} 1, \quad & \cos\left(\frac{2\pi i}{249}\right), \cos\left(\frac{2\pi i}{249}\right), \\ & \cos\left(\frac{4\pi i}{249}\right), \cos\left(\frac{4\pi i}{249}\right), \\ & \vdots \\ & \cos\left(\frac{248\pi i}{249}\right), \cos\left(\frac{248\pi i}{249}\right). \end{aligned}$$

That is, all but the largest one come in pairs of two equal eigenvalues. The eigenvectors corresponding to $\cos\left(\frac{2k\pi i}{249}\right)$, for $k > 0$ are

$$\begin{pmatrix} \cos\left(\frac{2\pi k}{249}\right) \\ \cos\left(\frac{4\pi k}{249}\right) \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} \sin\left(\frac{2\pi k}{249}\right) \\ \sin\left(\frac{4\pi k}{249}\right) \\ \vdots \\ 1 \end{pmatrix}$$

The eigenvector corresponding to the largest eigenvalue 1 is $(1, 1, \ldots, 1)^T$. Note that all A_i have the same eigenvectors. Since A is a positive linear combination of the A_i, the k^{th} largest eigenvalue of A is the same linear combination of the k^{th} largest eigenvalues of the A_i. That is, the k^{th} largest eigenvalue of A is

$$1 + \sum_{i=1}^{249} (.9)^i \cos\left(\frac{2\lfloor \frac{k}{2} \rfloor \pi i}{249}\right).$$

In particular, the first three eigenvalues are approximately:

18.99999999992728,

18.73270191995797,

and

18.73270191995797.

The first two values are fairly close together, their ratio being approximately 0.9859. Even the fourth and fifth eigenvalues are large, being approximately 0.9460 times as large as the first one.

5.3 The Test Cases

We performed asynchronous iterations in double precision, computing both the vector $\mathbf{x}^j$ and the approximate eigenvalue c_j until the following convergence criterion was met:

> *Convergence criterion:* Wait until every coordinate has been updated at least once and stop as soon as both of the following two conditions are met:
>
> - $\frac{|c_j - c_{j-1}|}{\min\{|c_j|, |c_{j-1}|\}} \leq 10^{-16}$
> - $\frac{\sqrt{\sum_{i=1}^{n} (c_j y_i^{j-1} - x_i^j)^2}}{\min\{|c_j|, |c_{j-1}|\}} \leq 10^{-16}$,
>
> where $y_i^k = x_i^k / \max\{|x_1^k|, \ldots, |x_n^k|\}$.

The first condition insures that the approximate eigenvalue has not changed much and the second insures that the product $A\mathbf{x}^{j-1}$ is approximately $c_j \mathbf{x}^{j-1}$.

The test cases described here are the same cases used in the videotape, however they are not the same runs described there. The reason is that there is a significant amount of time required, during the run, to write the information used in the video tape. The more processors used in a run, the more iterations were done, and the more writing that was done. The timings would not be indicative of the savings achieved by multiple processors if we timed the writing of the videotape information. Because the runs are not the same and the environment is stochastic, the numbers of iterations will be different also.

5.3.1 Synchronous Computation

We used a sequential vector-wise Jacobi scheme starting with $\mathbf{x}^0$ being a vector of numbers between 0 and 1, each chosen by a uniform pseudo-random number generator. The convergence criterion was met after 2332 iterations and 8hr 39min of wall-clock time on a single VAXstation 2000 dedicated to the task.

5.3.2 Asynchronous Computation

For the asynchronous computations, we divided the vector into nearly equal subvectors and updated

one subvector per iteration. We used both a fixed allocation and a cyclic allocation. In the fixed allocation, one processor was devoted to each block of coordinates, while in the cyclic allocation, whenever a processor required another block, it was assigned whichever block was next in the cycle.

Fixed Allocation We used an asynchronous fixed block-wise scheme with $k = 5$ blocks, 4 of size 100 and one of size 99. After 2hrs 15min and 23155 iterations, the convergence criterion was met. The five processors were not identical. The third one was a VAXstation 3200 and the other four were VAXstation 2000s. The fifth processor was busy with other work (unrelated to our calculations) to a greater extent than the other four. The numbers of iterations performed by each of the five processors were

3753 3903 9413 3782 2304.

Notice that the smallest number of iterations is about the same as the number of iterations required by the Jacobi vector-wise allocation. The starting vector $\mathbf{x}^0$ for this calculation was a unit vector with 1 in the first coordinate and 0 elsewhere.

Cyclic Allocation We used an asynchronous cyclic block-wise scheme with $k = 10$ blocks, 9 of size 50 and one of size 49. The starting vector $\mathbf{x}^0$ for this calculation was the same uniform pseudo-random vector used in the Jacobi scheme. After 1hr 9min and 28818 iterations, the convergence criterion was met. The 10 processors were all VAXstation 2000s or VAXstation IIs and the numbers of iterations performed on each of the 10 blocks were

2884 2877 2890 2888 2881
2890 2881 2888 2877 2865.

Since the 10 blocks were assigned to iterations cyclicly, we did not keep track of how many iterations were performed by each processor, but rather how many iterations updated the coordinates in each block. Notice that the 10 blocks were all updated approximately the same number of times when convergence occurred. There are two reasons why the numbers are not all equal. Most obvious is that there are still iterations ongoing when the convergence criterion is met. This, however, would not account for a difference of 35 iterations between two blocks. Such a difference is due to the nature of the asynchronous updating. Suppose a process finishes iteration k and begins iteration j. If this processor was particularly slow on this iteration, it may be that, for $i \in L_k$, $k < s_i^j$. That is, some other processor updated the coordinates in block L_k before the one which just finished, and the iteration which just finished must be ignored (otherwise it might be a "downdating" rather than an updating).

6 Animated Videotape

In the videotape we display the sequence of iterations for three different iterative schemes for finding the largest eigenvalue and eigenvector of a matrix A described earlier.

6.1 The Video System

A 512 × 512 × 8 pixel video frame buffer is installed in a VAX workstation to generate an RGB video signal under program control. The signal is translated by an encoder to NTSC video; NTSC is the United States standard for home television. The NTSC video signal is recorded on a 3/4 inch Umatic VCR. This VCR has the capability to edit single frames of video onto the tape under the direction of a controller in an IBM PC/XT which follows commands generated by a program running on the VAX.

The crucial point in the application of this system to the generation of video tapes is that the computations involved in generating a video image are quite separate from the actual recording of the video tape. A typical recording cycle requires about ten seconds to record a single video frame because of the time required to position the tape in the VCR. On the one hand this means that it takes a long time to generate a video tape (about 5 hours per minute of completed tape). On the other hand it makes a clear separation between the calculations needed to generate the image and the actual event of recording it. This allows fairly massive computations to be involved in the generation of the images without imposing the visual time-lag in viewing the resulting pictures.

6.2 Description of the Animation

Figure 1 exhibits output from a laser printer which shows what a single frame of the video tape looks like. This single frame illustrates the values of the components of a particular iterate. There is a bitmap which is 512 × 256 pixels. Each of the 512 columns is used to display a number. The 256 rows are divided into 64 groups of four pixels each. Each of the 64 groups is used to display the value of a single bit of the number in that column; all four pixels within the group have the same color. A double precision floating point number lives in 64 bits. On the VAX where this was done eight of the 64 bits are reserved for exponent and are ignored. The remaining 56 bits

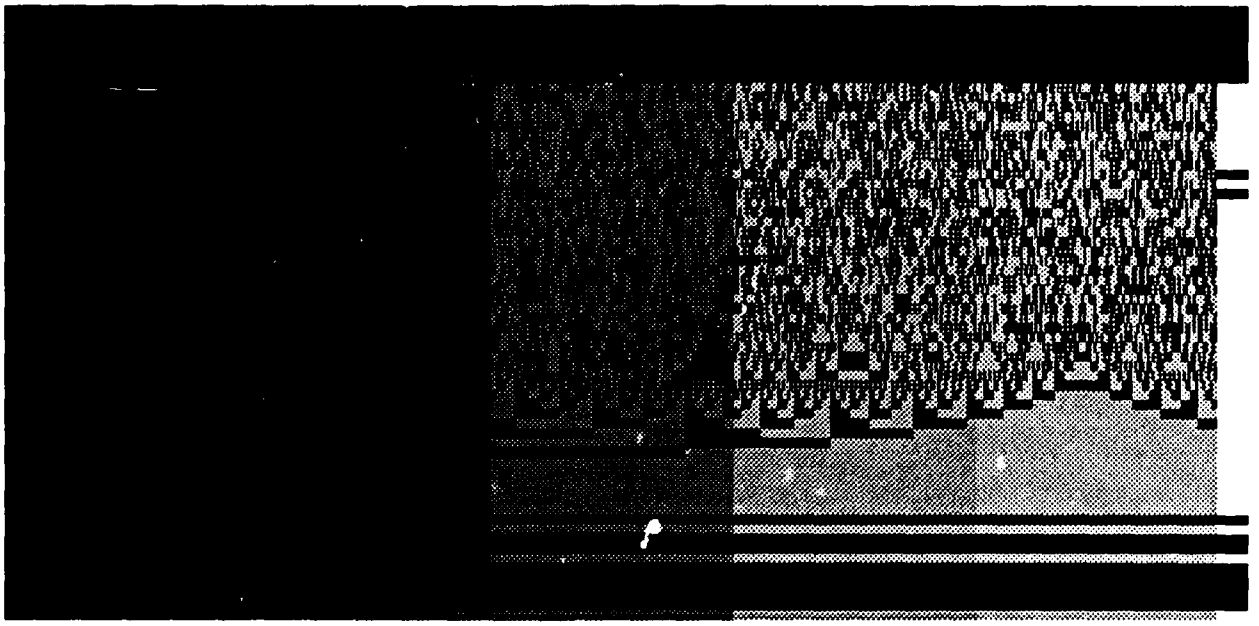


Figure 1: One Iteration From Eigenvalue Calculation, Cyclic Assignment

(of the fraction) are displayed with the most significant bit at the bottom and the least significant bit at the top. Figure 1 is from a fixed-blockwise evaluation, and the five shaded regions correspond to the five blocks and the five respective processors.

In order to understand this more precisely, look at the last 13 columns on the right of the bitmap. Using white for 1 and black for zero these thirteen column display the value

1042.99999999992728

The 56 bits in the floating point representation of this number are

10000010010111111111111111111
11111011011111111110000000000

In the videotape one should read bottom-to-top and white stands for 1. The reason the last 10 bits are zeros is that we added 1024 to the value 18.99999... to guarantee that all number had the same exponent.

Before viewing the actual video tape one anticipates (because of the standard theory of convergence for this calculation) seeing a "wave" of convergence sweep from bottom to top of the bitmap as the sequences of iterates converges to the solution. The video tape exhibits exactly this convergence. However, and this is what is important about the example, one also notices a number of additional features. First, there is a "cusp" in the convergence; in Figure 1 this cusp appears approximately 50 columns (10% percent of the bitmap) from the left edge. There is also an "anitcusp" approximately 300 columns (60% percent of the bitmap) from the left edge. Standard theory does not adequately explain the presence of these features although they are clearly related to the eigenvectors associated with the second largest (and smaller) eigenvalues. Second, there is an additional effect which is also visible in Figure 1 but is more pronounced in the animated sequence. Approximately four bits above the "zone of convergence" there is a very high frequency band of alternating bits. The "buzzing" of this band is very distinctive visually in the video tape and has no explanation known to us.

Acknowledgment

This research was supported in part by Office of Naval Research contract N00014-88-K0013 and by National Science Foundation grant DMS-8704218. The computer equipment used to perform the computations discussed herein was acquired with partial financial support from the National Science Foundation (under grants MCS 82-05126 and DMS 84-04927) and the Office of Naval Research (under contract N00014-84-G-0137). The video tapes were produced at the Pittsburgh Supercomputer Center with the assistance of Joel Welling.

7 References

Anderson, T.W. (1971). *The Statistical Analysis of Time Series.* John Wiley and Sons, New York.

Baudet, G.M. (1975). Asynchronous iterative methods for multiprocessors. *J. Assoc. Comput. Mach.*, **25**, 226-244.

Chazan, D. and Miranker, W. (1969). Chaotic relaxation. *Linear Algebra and Its Applications*, **2**, 199-222.

Eddy, W.F. and Schervish, M.J. (1986). Discrete-finite inference on a network of VAXes. *Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface*, 30-36. American Statistical Association, Washington, D.C.

Eddy, W.F. and Schervish, M.J. (1987). Parallel computing on a network of Vaxes with applications. *Proceedings of the Statistical Computing Section, American Statistical Association*, 41-47.

Lubachevsky, B., and Mitra, D. (1986). A chaotic, asynchronous algorithm for computing the fixed point of a nonnegative matrix of unit spectral radius. *J. Assoc. Comput. Mach.*, **33**, 130-150.

Schervish, M.J. (1988). Applications of parallel computation to statistical inference. *J. Amer. Statist. Assoc.*, **83**, (to appear).

Continuous Valued Neural Networks: Approximation Theoretic Results

George Cybenko
Center for Supercomputing Research and Development
and
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

We discuss results relevant to a class of neural networks that have close relationship to existing techniques in applied statistics such as density estimation, CART and projection pursuit. The perspective of this presentation is from that of approximation theory. We indicate how some statistical methods might be used to shed light on the behavior of neural networks.

1. Introduction

Neural computing is a general approach to computation that strives to use networks of simple processing elements instead of traditional procedural algorithms to implement a desired functional input/output relationship. Although the foundations of neural computation go back over thirty years, there was a long period in the 1970's during which interest in the technology dwindled partly because of some mathematically demonstrable limitations described by Minsky and Papert in [Mi69]. By the 1980's, researchers became confident that some of the limitations described by Minsky and Papert might be circumvented by making the underlying neural networks more complex (see for example [PDP,Ho82, Ho85]).

There are two major ways in which networks have been embellished to make them more powerful. One involves the introduction of feedback or stochastic mechanisms into the networks thereby making them dynamical systems capable of more complex behavior. The other, on which we focus in this paper, is the use of multilayered networks with theoretically unbounded order in the sense of [Mi69]. A fundamental advance with respect to multilayered networks has been the discovery of training algorithms that have worked well empirically in many applications [PDP].

This paper addresses a number of problems related to *multilayered, feedforward, continuous* (MFC) networks. We emphasize this restriction because many different ideas fall under the general rubric of neural network theory and, while we acknowledge the existence of other technologies (such as networks with feedback, various associative memories, Hopfield-Tank optimization networks, Boltzmann machines, etc.), we cannot pretend to deal with them all. Moreover, we believe the class of MFC networks to be the most promising for time series and other statistical applications. Indeed, the classical work of Widrow on adaptive filtering [Wi62] is perhaps the simplest manifestation of feedforward networks applied to statistical filtering problems. Needless to say, those ideas have proven to be extremely useful in applications such as channel equalization and echo cancelling in real-time telecommunications settings [Wi85]. More recently, there have been some interesting empirical studies done in nonlinear time series prediction that indicate some potential utility of neural networks in such an application [La87, Mo88].

We discuss some practical issues surrounding multilayered, feedforward, continuous networks especially in the context of known statistical techniques. The first question to be discussed concerns identifying the class of problems that can in principle be solved by MFC networks. On that point, we have obtained general results demonstrating that, at

This research was partially supported by Office of Naval Research grant N00014-87-K-0182 and National Science Foundation grant DCR-8619103.

least theoretically, networks with single internal hidden layers can be used to solve any continuous approximation problem [Cy88a, Cy88b]. Next, we discuss the class of problems that are feasibly (as opposed to theoretically) solvable by MFC networks. Finally, we discuss procedures for determining whether a candidate problem (as presented by empirical input/output data) might be feasibly solved by MFC networks.

In an area that is both promising and controversial, it is perhaps important to outline our perspective and philosophy on this general area of research. Our primary interest has been and continues to be the investigation of numerical algorithms for signal processing. We believe that MFC networks offer an interesting and potentially powerful technology for solving certain signal processing problems. At the same time, there are a number of known statistical techniques that share many basic ideas with MFC neural networks - namely, density estimation, CART and projection pursuit methods. We will attempt to bring some of these connections to light.

2. Technical Background

The neural networks of interest to us are multilayered feedforward continuous (MFC) networks. In order to discuss such networks, we introduce the notion of an N-node.

An N-node is a simple computational unit that accepts some number of real-valued inputs, applies an affine transformation to the inputs and then applies some fixed nonlinear function to this affine transformation. The output of an N-node is the output of the nonlinear function. In the sequel, we assume for simplicity that the nonlinearity is fixed for all nodes but that the affine transformations are of course node dependent. (The use of the same nonlinearity is arguably the most interesting case from an implementation point of view since then all nonlinear components are identical.)

Figure 1 graphically illustrates an N-node while the simple function that an N-node implements is given by

$$\sigma\left(\sum_{i=1}^{m} y_i x_i + \theta\right)$$

Here $X = (x_1, x_2, \ldots, x_m)$ are the real valued inputs to the node, $Y = (y_1, y_2, \ldots, y_m)$ are real valued constant weights, θ is a real constant and σ is some univariate function. The quantities $y_1, y_2, \ldots, y_m$ and θ determine the affine transformation at the node. An MFC network is built from such simple N-nodes by composition in layers.

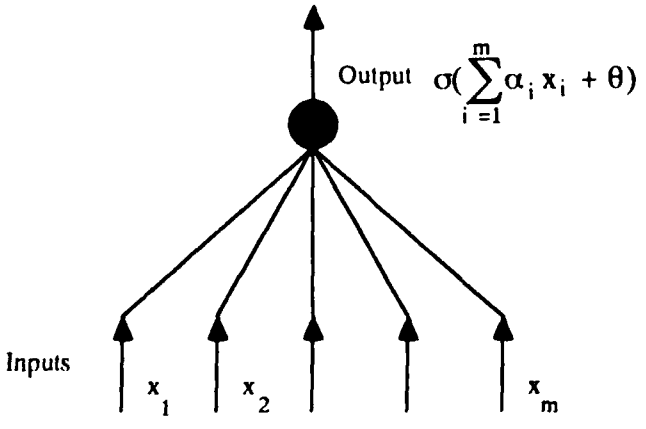


Figure 1.
Input-output relation of a single neural node

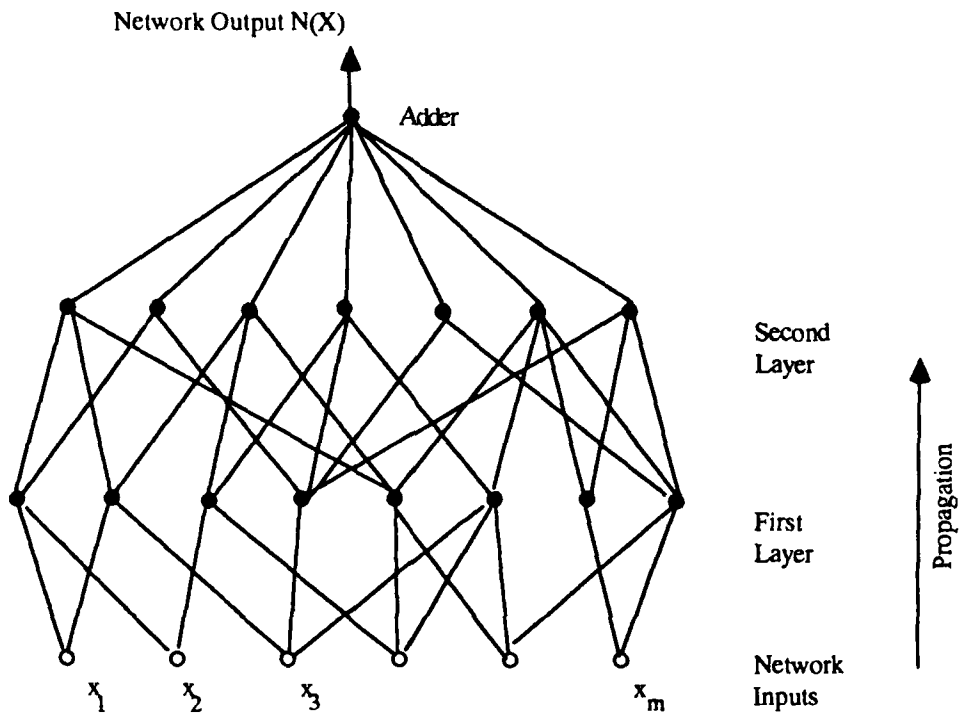


Figure 2
A sample network with two hidden layers

Figure 2 depicts a two layered MFC network. Generalizations to networks with more layers are done in the obvious manner. Without explicitly writing out the functional form of the network output, let us simply express the output as

$$N(X) = N(X,\Theta)$$

where Θ is a large dimensioned vector of the parameters in the network. These parameters include all weights and thresholds. Viewed as such, an MFC network implements one function from a family of functions parameterized by Θ.

Now suppose that some system produces samples of input/output data of the form

$$\{(X_i, f(X_i)),\ 1 \le i \le M\}.$$

Here f is the real-valued response function of the system - for input vector X, the system output is f(X). Based on these observations, an MFC network is sought that approximately interpolates the data and hopefully extrapolates to be a good approximation of f over the whole input domain of the system.

Thus we seek to find the parameters Θ that minimize some error criterion where the error is taken to be the difference between the actual system output and the network output.

Algorithms for adapting Θ to attempt to minimize this error criterion are called *supervised training, learning*, etc. algorithms. Viewing the situation from the perspective of nonlinear optimization, most of these learning algorithms are gradient descent methods whereby some estimate of the gradient of the error function (gradient with respect to the parameters Θ) is used to update and improve an estimate for Θ [Pa87].

Such empirical parametric model fitting is of course the essence of much of applied statistics and approximation theory and by no means a revolutionary idea in its own right. In fact, scientists and engineers have for centuries used parametric models such as polynomials, splines, rational functions, Fourier series, exponentials and so on to interpolate and extrapolate empirical data.

What then is the novelty of neural network theory? From the point of view of MFC networks, we believe that the novelty lies primarily in two quite different directions. First of all, the kinds of parametric models being used in neural network theory typically involve sigmoidal functions quite different and primitive in comparison with traditional algebraic or transcendental functions. The implications of using combinations and compositions of such primitive functions for approximation are not yet clear although sigmoidal functions that are normally used have certain locality properties that suggest robustness. Secondly, there is a preponderance of case studies and examples illustrating that MFC networks work reasonably well across an array of seemingly different applications. This is not to say that the MFC network approach is the best approach among many, just that it works quite often. In this respect, there is a certain similarity between MFC networks and simulated annealing [Ha85] - they both seem to be reasonably good at solving many different types of problems but for any given problem there may well be a better way to solve that problem. This fact alone begs for a better explanation.

There is of course another important innovation in that at some level of abstraction, MFC networks are biologically meaningful models of intelligent behavior and their study sheds light on the neurophysiological foundation of intelligence.

3. Theoretical Capabilities

For a given choice of network parameters, an MFC network implements a continuous function. Without constraining the architecture or size of the network, what kinds of functions can be arbitrarily well approximated by the output of a neural network? This of course depends heavily on the class of network architectures being considered and the type of nonlinearity implemented by a single node.

In prior research, we have definitively answered a number of these questions in a rigorous manner. Define a class of network architectures to be *complete* if: given a continuous function, f, with compact support and an $\varepsilon > 0$, there is a network from that class whose output approximates f uniformly to within ε over the support of f. For example, there are many classical classes of functions that are complete - polynomials, multinomials, Fourier series and so on.

We have shown that the following classes of networks are complete in this sense:

1. networks with two hidden, internal layers and any continuous sigmoidal nonlinearity [Cy88a];

2. networks with a single internal hidden layer and any continuous radial basis type function (see [Bu88,Ca87, Mo88,Po87] for discussions of radial basis functions - one can think of them as generalizations of spherically symmetric Gaussian densities in density estimation problems) as a nonlinearity [Cy88a];

3. networks with a single internal hidden layer and any continuous sigmoidal nonlinearity [Cy88b].

These results make absolutely no claims about the number of nodes needed to perform the approximation although in some cases, gross and probably unrealistic upper bounds could be obtained.

Of these three results, the last concerning networks with only one internal, hidden layer is certainly most surprising. It has generally been felt that such networks could implement decision functions for convex regions and there have been examples of special nonconvex regions being discriminated as well [Li87,Ni65,Wi87] but a general result has been missing. We believe that the results of

[Cy88b] are definitive in their resolution of the issue.

The proofs of 1. and 2. above are constructive and basically reduce to showing that networks in that class can implement so-called approximations to the identity or Parzen windows [Pa62,Du73] together with sums of such functions. It is well known that convolution with approximations to the identity approaches the identity function uniformly over a compact domain. What remains is to show that the convolution integrals can be uniformly approximated by finite Riemann sums over the whole domain. By contrast, the proof of 3. is nonconstructive, using the Hahn-Banach and Reisz Representation Theorems to show that a certain linear subspace is dense in the space of all continuous functions.

In summary, we feel that these results give rigorous meaning to the assertion that in principal any continuous function can be approximated by any of the three classes discussed above. Extensions to discontinuous, integrable functions are outlined in [Cy88b] as well.

Given that the classes of networks described above share the same completeness properties as many classical classes of functions (splines, polynomials, Fourier series, exponential families), what, if any, properties of MFC networks make them distinct? As we have mentioned before, in cases 1. and 2. above, the networks are capable of implementing Parzen window type estimators and hence there is a certain localization property that such approximations have. In a noisy approximation problem, this might be interpretable in terms of robustness. Secondly, the strong biological motivation makes the study of these types of approximating families interesting from a purely intellectual point of view - if indeed nature implements pattern recognition and classification this way using neurons, then it is interesting to understand how that is done.

4. Feasibility

The results summarized in the previous subsection indicate that network architectures and the nonlinearities that they implement do not constrain the kinds of problems that can be handled by MFC networks. However, in any real engineering attempt to implement a network solution, constraints must be imposed on the number of nodes used, the amount of data that can be observed and the complexity of the algorithm used to find suitable network parameters.
There have been numerous recent efforts trying to deal with such issues for a variety of different settings [Ah88,Ba88,Bl87, Ke87,Va84]. Valiant formalized a notion of feasibility with respect to learning a boolean function and demonstrated that certain classes of boolean functions were feasibly learnable in that sense [Va84]. (It should be clarified that in the context of our prior discussion, *learning* is any technique for selecting model parameters that let the parameterized system duplicate or approximate the input/output behavior of the observed system.) *Valiant* introduces a probabilistic setting for learning that is reminiscent of classical hypothesis testing.

Blumer et al. generalized Valiant's ideas to more general notions of learning (for example, learning rectangles or convex sets) and related feasibility in learning to the concept of Vapnik-Chervonenkis dimension in a nontrivial manner [Bl87]. Vapnik-Chervonenkis dimension was an idea introduced in nonparametric, distribution free pattern recognition some time ago [Va71] and its interpretation and utility in the context of learning is therefore quite natural although not at all obvious. Baum and Haussler have recently applied those results to neural networks with hard limiting nonlinearities by estimating the Vapnik-Chervonenkis dimension of a simple class of neural networks [Ba88]. However, the results of [Ba88] are disappointing from a practical point of view since the results make statements about the extent to which neural networks can accurately generalize assuming that some fraction of the empirical data presented to the network can be correctly *learned*, without directly addressing the difficult question of what sets of data can be learned by such (finite) networks. Recent work by Judd and Rivest [Ju88,Ri88] demonstrates that this is indeed a difficult question by showing that the problem of de-

termining whether a given network architecture can exactly implement a given empirical data set is in general NP-complete.

All of the research discussed above deals with Boolean (0,1) valued systems such as Boolean expressions and characteristic functions of sets. The situation with respect to real-valued functions and real-valued networks is largely uncharted territory. There has been some recent theoretical analysis of so-called universal Donsker classes [Du84,Du87] that generalize Vapnik-Chervonenkis classes in the context of distribution free limit theorems but even then, it appears that most interesting examples are closely related to the idea of Vapnik-Chervonenkis dimension anyway. There are some intriguing relationships between Donsker classes and metric entropy [Du87] that might be interpretable in terms of signal bandwidth - we discuss this shortly. Accordingly, most of the work on real-valued networks has been empirical (such as [La87,Mo88] and many papers in [NN1,NN2]).

The problem of approximating a real-valued function by some parametric combination and composition of simple functions is of course the raison d'etre of classical approximation theory. The traditional measure of how easy or hard a continuous function is to approximate is given by the magnitude of the function's derivative. Generally speaking, functions with small derivatives are easier to approximate because they change at a slower rate. However, even functions with small derivatives are very hard to approximate if the dimension of the underlying space is moderately large. A precise statement of this fact can be stated as follows:

Suppose that $|f(x)| < 1$,

$$\left| \frac{\partial f}{\partial x_i}(x) \right| < 1$$

for $x \in I_n$ (I_n being the unit n-cube in $\mathbf{R}^n$). Then if we seek an approximation g(x) so that $|f-g| < \varepsilon$ on I_n, we must sample f at more than $c\varepsilon^{-n}$ points for some constant c. Conversely, if chosen properly, $O(\varepsilon^{-n})$ points are sufficient.

A simple application of the mean value theorem shows that sampling f at that many points (properly distributed) is sufficient while constructing a simple class of functions f that oscillate unpredictably but within the constraints shows that that sampling is necessary. (The details are simple and the reader can easily fill them in.) For example, if we want to approximate such a function so that the approximation has two significant digits, then $\varepsilon = 0.01$ and for $n=6$ we need about 10^{12} samples of the function. This observation is completely independent of the technique that we use for approximating, be it polynomials, Fourier series or neural networks. Moreover, this could also be interpreted in terms of the classical sampling theory of multidimensional signal processing - signal bandwidth and the sampling rate are closely related in a like manner.

This example illustrates that smoothness of a function is not sufficient for making the problem of approximating the function feasible - the problem lies with the volume of the sample space as a function of linear dimension which grows exponentially in the number of variables. Accordingly, multidimensional approximation theory has largely restricted itself to problems involving very small dimensioned coordinate spaces. Similarly, empirical data analysis has had to be restricted to small dimensions. Two notable exceptions are the techniques of *projection pursuit* [Hu85] and *CART* (classification and regression trees) [Br84].

One of the guiding principles of both neural network theory and projection pursuit methods is that some multidimensional functions have parsimonious representations in terms of linear combinations and functions of a single variable. Linear combinations and univariate functions are considered relatively easy to estimate and compute. That attitude is encouraged by the well-known result of

Kolmogorov [Ko57,Lo76] that goes as follows:

Theorem [Kolmogorov] There exist m(2m+1) continuous increasing univariate functions h_{pq} with the property that given any continuous function f on I_m, there is a continuous univariate function g so that

$$f(x_1, \ldots, x_m) = \sum_{q=1}^{2m+1} g\left(\sum_{p=1}^{m} h_{pq}(x_p) \right)$$

This representation involves summations, fixed univariate functions and only one univariate function that is not predetermined, namely g(x). While superficially this sounds encouraging, it packs all of the complexity of the multidimensional function f into the univariate function g. See [Di84] for some discussion of the properties of functions representable in such terms involving polynomials only.

We have tried to investigate the Kolmogorov function, g, defined above for a complex problem in spectral estimation. Our numerical experiments sought to get least squares estimates of g with increasing accuracy. The results clearly show that the complexity of g is enormous - it a highly oscillatory function that is poorly approximated by Fourier series or other orthogonal basis functions. This leads us to conjecture the existence of a relationship between the complexity of a general multidimensional function, f, and the complexity of its univariate version, g, via the Kolmogorov representation. The complexity of a function can be measured for instance in terms of its bandwidth (ie spectrum). We believe that there are severe limitations on the *complexity of multidimensional functions* that can be implemented as simple combinations and compositions of univariate functions such as sigmoidals. We believe that some research ought to be devoted to such questions.

At the same time as we outline this dismal situation, there have been a number of examples where MFC networks have done an admirable job of modeling and approximating complex time series via nonlinear prediction [La87,Mo88]. Those examples require a closer look to see exactly what kind of mechanism is used for generating the time series. The example in [La87] shows that a simple network can learn and then replicate quite well the behavior of the quadratic map of the unit interval into itself given by f(x,b) = bx(1-x). The time series is generated by iterating f. This family of maps, as b varies, exhibits period doubling and chaos. Hence, for different values of b, a plot of the time series can look impressively complex. However, the underlying function itself that generates this complex behavior is by any measure very simple to approximate. It is a two dimensional quadratic function. The general theory outlined by Feigenbaum [Fe78] shows that the behavior exhibited by bx(1-x) is generic and will be exhibited by any function that is unimodal with a quadratic maximum. Hence, any reasonable approximation would likely have similar behavior.

The time series modeled in [Mo88] is generated by the Mackey-Glass equation which is a more complex example of chaotic behavior. Nonetheless, the model used four prior samples of the series to predict, nonlinearly, a future sample. The modeling in [Mo88] basically involves estimating a real-valued function of four real variables and, by our previous observations, this comes close to what must be regarded as a feasible problem to solve in general. To understand this particular example better, we need to examine solutions to the Mackey-Glass equation and see if they possess any special properties, in terms of either predictability or smoothness.

5. Determining Feasibility

The discussion of the previous paragraph surrounded the question of identifying general analytic criteria for determining the feasibility of using MFC networks to implement approximate solutions to problems in continuous-valued applications. These applications include nonlinear time series predic-

tion and the implementation of difficult to compute functions.

The practical problem remains of deciding whether a given empirical data set is of the type that could be feasibly implemented by an MFC network. It may not be possible to determine whether the underlying application satisfies the requisite criteria, whatever they may be. This would be the case in, for example, continuous recognition problems such as signature classification from sonar, IR or radar imaging data. The underlying analytical model that determines the classification may be too complex or difficult to express explicitly to decide whether the application can be well served by MFC networks.

In fact, classification using many continuous input variables is a general application area that has been successfully handled by MFC networks ([Se88] for example) in some cases. We introduce the informal notion of *granularity* as a parameter of an approximation problem in the following way: *granularity* refers to the number of distinct function values that are of interest in an MFC application - a finely grained problem is one that involves many function values over a large part of the input variable space while a coarse application is a problem with few function values of interest and the regions where the function assumes all except one of those values are sparsely distributed in the input space.

Thus in a classification problem involving the recognition of say 10 signatures from 20 real-valued signal statistics would be characterized as a coarse problem since the classification would be nontrivial typically in only 10 isolated regions of the 20 dimensioned input space. Thus, relatively speaking, the volume of the input space that involves an interesting function value is relatively small even though there are many real-valued input variables. Moreover, the precision sought in such a classification problem is relatively low compared with an application such as time series prediction. In a qualitative way, let a coarse problem be one that involves some combination of these features. What is an appropriate quantitative measure of granularity as discussed above?

In questions such as this, we believe that guidance must be sought from very similar kinds of problems studied by statisticians in the general methodology of CART (classification and regression trees) [Br84]. CART is a statistically based, data driven method for partitioning an empirical data set typically using a succession of linear discriminant functions. Loosely speaking, the hierarchy of linear discriminations determines a binary decision tree which is something very similar in fact to a multilayered neural network with hardlimiting nonlinearities. The technique of projection pursuit [Hu85] involves computing good (with respect to some criterion) projections of multidimensional data onto a vector direction and performing general nonlinear regression on the projected data. That basic step of projection and regression is iterated on the residual data. The resulting functional form of the approximation resembles the Kolmogorov representation very closely and this is discussed in more detail in [Di84].

We pose the following two questions as a challenge for the statistical audience, given the various observations that we have made above.

> Can statistical techniques such as CART and projection pursuit be used as preprocessing steps for determining the feasibility of applying MFC networks to a specific empirical data set?
>
> How does the performance of MFC networks compare with CART and projection pursuit on sparse continuous classification problems?

In summary, we believe there are valuable contributions to be made by using known statistical techniques to assess the feasibility of using MFC neural networks in a variety of problems.

Bibliography

[Ah88] S. Ahmad, "A study of scaling and generalization in neural networks",

University of Illinois - Urbana, Department of Computer Science, Tech. Rep. R-88-1454, 1988.

[Ba88] E. Baum and D. Haussler, "What size net gives valid generalization", *Neural Computation*, to appear.

[Bl86] A. Blumer, A. Ehrenfeucht, D. Haussler and M.K. Warmuth, "Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension", *Proceedings 18th ACM Symposium on Theory of Computation*, pp. 273-282, 1986.

[Br84] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, *Classification and Regression Trees*, Wadsworth International Group,Belmont, CA,1984.

[Bu88] M.D. Buhmann, "Multivariate interpolation in odd dimensional Euclidean spaces using multiquadratics", *University of Cambridge, Dept. of Appl. Math. and Theor. Physics, Tech. Rep. DAMTP 1988/NA6*, 1988.

[Ca87] M. Casdagli, "Nonlinear prediction of chaotic time series", *Physica D*, submitted 1987.

[Cy88a] G. Cybenko, "Continuous valued neural networks with two hidden layers are sufficient ",submitted for publication in *Mathematics of Control, Signals and Systems*, 1988.

[Cy88b] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, submitted 1988.

[Du73] R. Duda and P. Hart , *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[Du84] R.M. Dudley, *A course on empirical processes*, Lecture Notes in Mathematics No. 1097, Springer-Verlag,New York, 1984.

[Du87] R.M. Dudley, "Universal Donsker classes and metric entropy", *Ann.Prob.*, Vol. 15(4), pp. 1306-1326, 1987.

[Di84] P. Diaconis and M. Shahshanani, "On nonlinear functions of linear combinations" , *SIAM Journal on Scientific and Statistical Computing*, Vol. 5, pp. 175-191, 984.

[Fe78] M.J. Feigenbaum, "Quantitative universality for a class of nonlinear transformations", *Journal of Statistical Physics*, Vol. 19, pp. 25-52,1978.

[Ha85] B. Hajek, "A tutorial survey of theory and applications of simulated annealing", in *Proceedings of 24th Conference on Decision and Control*, Ft. Lauderdale, pp. 775-760, 1985.

[Ho82] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Science, USA*,1982, Vol. 79, pp. 2554-2558.

[Ho85] J.J. Hopfield and D.W. Tank, "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics*, 1985, pp. 141, Vol. 52.

[Hu85] P.J. Huber, "Projection pursuit", *Annals of Statistics*, Vol. 13, pp. 435-475, 1985.

[Ju88] J.S. Judd, "The intractability of learning in connectionist networks", *University of Massachusetts COINS Dept. Tech. Rep.*, March 1988.

[Ke87] M. Kearns, M. Li, L. Pitt and L. Valiant, "Recent results on Boolean concept learning", *Proc. 4th Int. Workshop on Machine Learning*, pp. 337-352, 1987.

[Ko57] A.N. Kolmogorov, *Dokl. Akad. Nauk. SSSR*, Vol. 114,pp. 953-956, 1957.

[La87] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modeling", Theoretical Division, Los Alamos National Laboratory, 1987.

[Li87] W.Y. Huang and R.P. Lippmann, "Neural net and traditional classifiers", MIT Lincoln Laboratory Technical Report, 1987.

[Lo76] G.G. Lorentz, "The 13th problem of Hilbert", in *Mathematical Developments Arising from Hilbert's Problems* (Edited by F. Browder), American Mathematical Society, Providence, RI, 1976.

[Mi69] M. Minsky and S. Papert, *Perceptrons*, MIT Press, 1969.

[Mo88] J. Moody and C. Darken, "Learning with localized receptive fields", *Yale University, Department*

of Computer Science , Tech. Rep. DCS/RR-649, September 1988.

[Ni65] N.J. Nilson, *Learning Machines*, McGraw Hill, New York, 1965.

[NN1] *Proceedings of IEEE First International Conference on Neural Networks*, San Diego,1987.

[NN2] *Proceedings of IEEE Second International Conference on Neural Networks*, San Diego, 1988.

[Pa62] E. Parzen, On estimation of a probability density and mode", *Ann. Math. Stat.* Vol. 33, pp.1065-1076, 1962.

[Pa87] D.B. Parker, "Optimal algorithms for adaptive networks", in *Proceedings of IEEE First International Conference on Neural Networks*, San Diego, pp. 593-600, 1987.

[Po87] M.J.D. Powell, "Radial basis functions for multivariable interpolation: a review", *IMA Conference on Algorithms for the Approximation of Functions and Data*, Oxford University Press, 1987.

[PDP] D.E. Rumelhart, G.E. Hinton and J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, MIT Press, 1986.

[Ri88] A. Blum and R. Rivest, "Training a 3-node neural network is NP-complete", *Proceedings First Workshop on Computational Learning Theory MIT*, 1988.

[Se88] R.P. Gorman and T.J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets", *Neural Networks*, Vol. 1, pp.75, 1988.

[Va71] V.N. Vapnik and A.Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities", *Th. Prob. and its Appl.*, Vol. 16(2), pp. 264-280, 1971.

[Va84] L.G. Valiant, "A Theory of the Learnable", *Communications of the ACM*, 1984, Vol. 27:11, pp. 1134-1142.

[Wi62] B. Widrow, "Generalization and Information Storage in Networks of Adaline Neurons", *Self-Organizing Systems 1962*, Spartan Books, Washington, 1962.

[Wi87] A. Wieland and R. Leighton, "Geometric analysis of neural network capabilities",in *Proceedings of 1st IEEE International Conference on Neural Networks*, San Diego, CA 1987.

[Wi85] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice-Hall , Englewood Cliffs, NJ, 1985.

Parameter Identification for Stochastic Neural Systems[1]

Muhammad K. Habib
Center for Computational Statistics and Probability
George Mason University
Fairfax, VA 22030

Abstract

Stochastic models of some aspects of the electrical activity in the nervous system at the cellular and network levels are investigated. In particular, models of the subthreshold activity of the somal transmembrane potential of neurons are considered along with methods of identification of physiological parameters of the discussed models. A simulation study is conducted to evaluate the performance and efficiency of the estimates of the parameters.

1. *Introduction.* Studies of mechanisms underlying neural coding and the representation of information in the nervous system are of great interest to neuroscientists and modelers of neural networks. Stochastic models are essential tools in describing the behavior of neurons under conditions where large numbers of inputs and internal events occur at the cellular and network levels. For instance, there is an extensive literature concerning experimental and theoretical studies of neuronal integration of synaptic inputs as reflected by the difference in potential across the somal membrane of nerve cells (see e.g. Johannesma, 1968; Tuckwell, 1979; Ricciardi and Sacerdote, 1979; Baranyi and Feher, 1981; Kallianpur, 1983; Habib, 1985; Ferster, 1987; Habib and Thavaneswaran, 1988.) The stochastic models developed in some of these studies relate the subthreshold behavior of somal membrane potential near the spike generation (or initial) region to physiologically meaningful parameters. These include the effective membrane time constant, amplitudes and rate of occurrences of membrane perturbations due to the arrival of excitatory and inhibitory post-synaptic potentials (EPSPs and IPSPs, respectively), and measures of variability of synaptic inputs. Estimation of these parameters using experimentally generated intracellular recordings of the neuronal membrane potential should shed light on some aspects of neuronal integration of synapic input.

In Section 2, we present several Ito-type stochastic differential equation models that describe the activity of different types of neurons or activity of certain type of neurons under different experimental conditions. In Section 3, we discuss statistical methods of parameter estimation such as maximum likelihood and the theory of optimal estimating functions. In Section 4, we report on a simulation study to evaluate the performance of the parameter estimators.

[1]This research was supported by research contract with the Office of Naval Research, Contract Number N00014-83-K-0387.

2. ***Stochastic Neuronal Models.*** Assume that the state of the neuron is characterized by the difference in potential across its membrane near a spatially restricted area of the soma called the trigger zone (or spike initiation region). The membrane potential is modeled by a stochastic process, V(t), defined on a probability space $(\Omega, \mathcal{F}, P)$. It is subject to instantaneous changes due to the occurrence of a) EPSPs which are assumed to occur according to mutually independent Poisson processes $P(\lambda_k^e; t)$ with rates λ_k^e $(k=1,2,\ldots,n_1)$, each accompanied by an instantaneous displacement of V(t) by a constant amount $\alpha_k^e > 0$ $(k=1,2,\ldots,n_1)$, and b) IPSP which occur according to independent Poisson processes $P(\lambda_k^i; t)$ with effective displacement $\alpha_k^i > 0$ ($k = 1, 2, \ldots, n_2$). Between PSPs, V(t) decays exponentially to a resting potential with time constant τ. As a first approximation the PSPs are assumed to sum linearly at the trigger zone, and when V(t) reaches the neuron's threshold, an action potential takes place. Following the action potential, V(t) is reset to a resting potential. Based on this simplified model neuron and considering n_1 excitatory synapses and n_2 inhibitory ones, the membrane potential V(t), is modeled as a solution of the stochastic differential equation

$$dV(t) = \rho V(t)dt + \sum_{k=1}^{n_1} \alpha_k^e \, dP(\lambda_k^e;t) - \sum_{k=1}^{n_2} \alpha_k^i \, dP(\lambda_k^i ;t), \tag{2.1}$$

where $V(0) = V_0$ and $\rho = \tau^{-1}$. Under certain conditions the solution of (2.1) is a homogeneous Markov process with discontinuous sample paths. This model is known as Stein's model (Stein, 1965) and is a special case of the well known Poisson driven Markov process models. This model has been treated in the literature by many authors, among them Johannesma (1968) and Tuckwell (1979).

Diffusion models in which the discontinuities of V(t) are smoothed out have been sought as approximations to the discontinuous model (2.1) (see e.g. Ricciardi, 1982; Kallianpur, 1983; Lansky and Lanska, 1987). These types of approximations are justified on the grounds that for many types of neurons in the central nervous system, synapses are densely packed along the dentritic tree. If the jumps of V(t) are small and the rates of occurrence of the post-synaptic potentials are very large, then the approximation of the Poisson driven Markov model by a diffusion model is appropriate and is accomplished by allowing the amplitudes α_k^e , α_k^i to tend to zero and the frequencies λ_k^e , λ_k^i to become large in a certain manner. Under some regularity conditions it was shown that model (2.1) can be approximated by the diffusion model

$$dV(t) = (-\rho V(t) + \mu)\, dt + \sigma \, dW(t), \; 0 < t < T, \tag{2.2}$$

$V(0) = V_0$, where W is the standard Wiener process (or Brownian motion).

As has been mentioned, model (2.2) describes the subthreshold activity of the somal membrane

potential of neurons which receive extensive (or rapid) synaptic input with relatively small potential displacements. This model may be suited for neurons which are spontaneously active. However, in many situations especially for stimulus driven neurons this last assumption on synaptic input might be too stringent because the nerve cell might receive a limited number of effective synaptic inputs that induce relatively large potential displacements, in addition to the extensive synaptic diffusion inputs discuss ed above. For example, in a study of the organization of inputs from the lateral geniculate nucleus to cells in the striate cortex of the cat, Tanaka (1983) found that about 10 genicular neurons are functionally connected to one simple-cell during the presentation of effective stimuli. A large (converence) number (more than 30) was obtained from studies of geniculate projection to complex cells. In this case a mixed model of diffusion and point process inputs may be more suitable for describing the activity of such cortical neurons. To that end, assume that in addition to the extensive synaptic input leading to the diffusion model (2.2), there are n_1 EPSPs arriving according to independent Poisson processes $N(\lambda_k^e, t)$ with random intensities λ_k^e, and EPSP displacement amplitudes α_k^e, $k=1,2,\ldots,n_1$. In addition, IPSPs are arriving according to the independent processes $N(\lambda_k^i, t)$, with the corresponding parameters λ_k^i and α_k^i, $k=1,2,\ldots,n_2$. This setup leads to the following extended mixed model to describe the membrane potential of a stimulus driven neuron:

$$(2.3) \qquad dV(t) = (-\rho V(t) + \mu)\, dt + \sigma\, dW(t) + \sum_{k=1}^{n_1} \alpha_k^e dN(\lambda_k^e, t) - \sum_{k=1}^{n_2} \alpha_k^i dN(\lambda_k^i, t).$$

Model (2.3) is remarkably similar to the continuous neuronal model proposed by Hopfield (1984). The problem of parameter estimation of the mixed model has not been sufficiently addressed in the literature. In the next section we treat the problem of parameter estimation of the diffusion model (2.2) and the mixed model (2.3).

3. ***Parameter Estimation of a Diffusion Neuronal Model.*** Lansky (1983, 1984) considered the problem of parameter estimation for diffusion neuronal models observed over a fixed interval $[0,T]$ and discussed the asymptotic properties of the estimators as $T\to\infty$. Given n independent trajectories $\{V_k(t), 0 < t < \tau_k\}$ $k = 1, 2, \ldots, n$, where, $\tau_1, \tau_2, \ldots, \tau_n$ are independent random variables (stopping times) with $P(\tau_k < \infty) = 1$, $k = 1, 2, \ldots, n$.

Habib (1985) derived maximum likelihood estimators of the parameters ρ and μ and established their large sample properties such as strong consistency and asymptotic normality assuming σ is known. Now recall the diffusion neuronal model (2.2). From Sorensen (1983), the log-likelihood function is given by

$$(3.1)\qquad L_n(\rho,\mu)=\sum_{k=1}^{n}\{\int_{\tau_{k-1}}^{\tau_k}(-\rho V_k(t)+\mu)\,dV_k(t)-\tfrac{1}{2}\int_{\tau_{k-1}}^{\tau_k}(-\rho V_k(t)+\mu)^2\,dt\}.$$

The maximum likelihood estimators (MLE) $\hat{\rho}_n$ and $\hat{\mu}_n$ of ρ and μ respectively are simply those values given by

$$(3.2)\qquad \hat{\rho}_n=\frac{D_n[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k(t)\,dV_k(t)]-[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k(t)\,dt][\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}dV_k(t)]}{[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k(t)\,dt]^2-D_nE_n}$$

$$(3.3)\qquad \hat{\mu}_n=\frac{[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k(t)\,dt][\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k(t)\,dV_k(t)-E_n[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}dV_k(t)]}{[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k(t)\,dt]^2-D_nE_n}$$

where

$$D_n=[\sum_{k=1}^{n}(\tau_k-\tau_{k-1})]\quad\text{and}\quad E_n=[\sum_{k=1}^{n}\int_{\tau_{k-1}}^{\tau_k}V_k^2(t)\,dt].$$

Using the fact that the membrane potential $V_k(t)$ is observed continuously over random intervals, the diffusion coefficient s may be estimated from an observed trajectory V_k (k=1,2,...,n) by the formula

$$(3.4)\qquad \hat{\sigma}^2(k)=\frac{1}{(\tau_k-\tau_{k-1})}\lim_{m\to\infty}\sum_{j=1}^{2^{m_k}}[V_k(\tau_{k-1}+jd_k2^{-m_k})-V_k(\tau_{k-1}+(j-1)d_k2^{-m_k})]^2.$$

This result may be proved using the corresponding result of Levy for Brownian motion by transforming V_k via time substitutions into Brownian motion (or Wiener process). A natural estimate of σ^2 which employs all the observed trajectories is given by

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{k=1}^{n} \hat{\sigma}^2(k).$$

The consistency and asymptotic normality of $\hat{\rho}_n$ and $\hat{\mu}_n$ (as $n\to\infty$) have been established in Habib (1985).

4. *Simulation Studies.* In this section we briefly discuss the results of a simulation study to evaluate the performance and efficiency of estimates of the parameters ρ and μ of model (2.2). This study provides general guidelines for the choice of the number of observed trajectories and the length of the observation period of every trajectory.

For simplicity, we consider the diffusion model (2.2). Assume for the moment, that the period of observation is fixed, say [0,T]. In this case, the estimators $\hat{\rho}_{n,T}$ and $\hat{\mu}_{n,T}$ are defined in terms of stochastic and ordinary integrals (c.f. (3.2) and (3.3)). But, in practice one has to approximate these integrals with appropriate finite sums which depend on the digitization scheme or the partition mesh $\{t_0,t_1,\ldots,t_K\} \subset [0,T]$.

In order to evaluate the performance of the estimates $\hat{\rho}_{n,T}$ and $\hat{\mu}_{n,T}$, we simulated the solution of model (2.2) using the difference equation

$$V(t_{k+1}) = (-\rho V(t_k) + \mu)\, h + \sigma(W(t_{k+1}) - W(t_k)) \tag{4.1}$$

where $h = t/K$, $t_k = kh$, $k=1,2,\ldots,K$. It is well known that the solution of (2.8) converges to V(t). For instance, if we set $V_K(t) = V(t_k)$ for $t\varepsilon[t_k, t_{k+1})$, then

$$E\Big(\sup_{0\le t\le T} |\, V(t) - V_K(t)\,|^2\Big) \to 0.$$

as $K\to\infty$ (see Gihman and Skorokhod, 1979). This and other kinds of discretization, especially Runga-Kutta schemes, have been extensively studied (see e.g. Magshoodi and Harris, 1987).

It is clear from Table 4.1 that for processes which are observed over a period [0,T) with T=10

ms, the estimates of all parameters except for σ are very close to the true values of the parameters and they improve as the number of observed trajectories, n, increases. From Table 4.2, there is no improvement in the estimators as the number of observed trajectories n increases (in fact, they deteriorate). This apparently happens because for Table 4.2 the period of observation [0,T] was longer, T = 15 ms. Therefore, one may conclude that for action potentials with long durations, one does not gain much by recording a large number of spikes, but for action potentials with relatively short durations, one can expect that the parameter estimators will improve as the number of observed action potentials increases.

5. *Conclusions.* The stochastic models considered in Section 2, take into account only the temporal aspects of synaptic input. It is well established, though, that among the important factors influencing synaptic integration are the geometry of the dendrites of post-synaptic neurons and the spatial organization of synaptic input. Habib and Thavaneswaran (1988) proposed a stochastic partial differential equation which is based on a cable model of a system of branched dendrites projected onto a one dimensional equivalent dendrite as proposed by Rall (1978). The theory of optimal estimating functions was applied in this case to obtain estimates of the model's parameters.

Table 4.1 : Parameter estimates using a simulated diffusion process observed n-times over a fixed period [0,T] and sampled every σ units:

a. T = 10 m.s., δ = 0.10.

Parameters	True Value	Estimated Value n=1	Estimated Value n=10	Estimated Value n=50
$\rho = \tau^{-1}$	0.33333	0.30336	0.33000	0.33427
μ	5.00000	4.63803	4.84648	4.88702
σ	0.31623	0.67566	0.67364	0.67583

Table 4.2 : Parameter estimates using a simulated diffusion process observed n-times over a fixed period [0,T] and sampled every δ units:

b. T = 20 m.s., δ = 0.10.

Parameters	True Value	Estimated Value n=1	Estimated Value n=10	Estimated Value n=50
$\rho = \tau^{-1}$	0.33333	0.30369	0.32705	0.32399
μ	5.00000	4.86121	4.77822	4.71001
σ	0.31623	0.33012	0.51796	0.33537

Before concluding, it should be noted that the parameters ρ and μ in the mixed Ito-Markov model (2.3) may be estimated using the theory of optimal estimating functions. Indeed, let

$$N(t) = \sum_{k=1}^{n_1} \alpha_k^e \, N(\lambda_k^e, t) - \sum_{k=1}^{n_2} \alpha_k^i \, N(\lambda_k^i, t), \tag{5.1}$$

and

$$E[N(t)] = \left(\sum_{k=1}^{n_1} \alpha_k^e \, \lambda_k^e - \sum_{k=1}^{n_2} \alpha_k^i \, \lambda_k^i\right) t = \lambda t. \tag{5.2}$$

Notice that $M(t) = W(t) + N(t) - \lambda t$ is a martingale with $M(0) = 0$. Substituting in (2.3), we obtain the equivalent model:

$$dV(t) = (-\rho V(t) + \mu' t)\, dt + dM(t), \tag{5.3}$$

where $\mu' = \mu + \lambda$. The method of optimal estimating functions can be used in this case and it can be shown that the optimal estimates of ρ and μ are identical to the maximum likelihood estimates $\hat{\rho}$ and $\hat{\mu}$ in (3.2) and (3.3).

One may then estimate the parameters ρ and μ of model (2.2) from data recorded while the neuron is spontaneously active. In the meantime, the parameters ρ and μ' of model (5.3) may be estimated form data recorded from the same neuron during periods of stimulus-driven activity. In this case, it is possible to estimate the parameter $\lambda = \mu' - \mu$ which reflects the impact of the synaptic activity due to the presence of the stimulus. Also a change in the value of the parameter ρ may reflect changes in the membrane properties due to the stimulus.

REFERENCES

Baranyi, A. and Feher, O. (1981) Intracellular studies of cortical synaptic placticity. *Exp. Brain. Res* 41: 124-134.

Ferster, D. (1987) Origin of orientation selective EPSPs in neurons of cat visual cortex. *J. Neurosci.* 7: 1780-1791.

Gihman, I.I. and Skorokhod, A.V. (1979) *Stochastic Differential Equations*, Springer-Verlag, New York.

Habib, M.K. (1985) Parameter estimation for randomly stopped processess and neuronal modeling. UNC Institute of Statistics, Mimeo Series No. 1492.

Habib, M.K. and Thavaneswaran, A. (1988) Optimal Estimation for Semimartingale Neuronal Models. *Proceeding of a conference on Stochastic Methods for Biological Intelligence*: Editors M.K. Habib and

J. Davis. Plenum Publishers.

Hopfield, J. J. (1984) Neurons with graded response have collective properties like those of two-slate neurons. *Proc. Math. Acad. Sci.* 80: 3088-3092.

Johannesma, P. I. M. (1968) Diffusion models for the stochastic activity of neurons. In Caianello, E.R., (ed.) *Neuronal Networks.* New York, Springer Verlag.

Kallianpur, G. (1983) On the diffusion approximation to a discontinuous model for a single neuron. In: Sen P.K. (ed) *Contributions to Statistics.* North-Holland, Amsterdam.

Lansky, P. (1983) Inference for the diffusion models of neuronal activitv *Math. Bioscien.* 67: 247-260.

Lansky, P. (1984) On approximations of Stein's neuronal model. *J. Theor. Biol.* 107: 631-647.

Lansky, P. and Lanska, V. (1987) Diffusion approximation of the neuronal model with synaptic reverse potentials. *Biol. Cybern.* 56: 19-26.

Magshoodi, Y. and Harris, C. J. (1987) On probability approximation and simulation of non-linear jump-diffusion stochastic differential equations. *IMA J. Math. Cont. Inform.* 1: 1-28.

Rall, W. (1978) Core conductor theory and cable properties of neurons. *Handbook of Physiology - The Nervous System.* I. Vol. 1. Amer. Physiolog. Soc., Bethesda, MD.

Ricciardi, L. M. (1982) Diffusion approximations and computational problems for single neurons activity. In: Amari S., Arbib M. A. (eds) *Competition and cooperation in neural networks.* Lecture Notes in Biomathematics 45: 143-154.

Ricciardi, L. M. and Sacerdote, L (1979) The Ornstein-Uhlenbeck process as a model for neuronal activity. *Biological Cybernet.* 35: 1-9.

Sorensen, L. M. (1983) On maximum likelihood estimation in randomly stopped diffusion type processes. *Intern. Statist. Rev.* 51: 93-110.

Stein, R. B. (1965) Some models of neuronal variability. *Biophysical J.* 5: 173-195.

Tanaka, K. (1983) Cross-correlation analysis of geniculastriate neuronal relationships in cats. *J. Neurophysiol.* 49: 1303-1318.

Tuckwell, H. C. (1979) Synaptic transmission in a model for stochastic neural activity. *J. Theor. Biol.* 77: 65-81.

STATISTICAL LEARNING NETWORKS: A UNIFYING VIEW

Andrew R. Barron[1], University of Illinois
Roger L. Barron, Barron Associates, Inc.

Abstract

A variety of network models for empirical inference have been introduced in rudimentary form as models for neurological computation. Motivated in part by these brain models and to a greater extent motivated by the need for general purpose capabilities for empirical estimation and classification, learning network models have been developed and successfully applied to complex engineering problems for at least 25 years. In the statistics community, there is considerable interest in similar models for the inference of high-dimensional relationships. In these methods, functions of many variables are estimated by composing functions of more tractable lower-dimensional forms. In this presentation, we describe the commonality as well as the diversity of the network models introduced in these different settings and point toward some new developments.

1. Introduction

In the context of empirical inference of functions of many variables, a *network* is a function represented by the composition of many basic functions. The basic functions (which are also called elements, units, building blocks, network nodes, or sometimes artificial neurons) are constrained in form: typically nonlinear functions of a few variables or linear functions of many variables. By definition, a *learning network* estimates its function from representative observations of the relevant variables.

Several composition schemes for network functions and corresponding estimation algorithms are reviewed in this paper. Consideration is given to certain networks popular in the neurocomputing field such as perceptrons, madelines, and backpropagation networks. (For a collection of some of the key papers in this field see the volume edited by Anderson and Rosenfeld 1988.) Unfortunately many learning networks are inflexible in the form of the basic functions, inflexible in the connectivity of the network, and lack global optimization of the network function. More consideration is given here to globally optimized networks, networks with adaptively synthesized structure, and networks with nonparametrically estimated units. Particular attention is given to polynomial networks (R.L. Barron et al. 1964, 1975, 1984, Ivakhnenko 1971), projection pursuit (Friedman et al. 1974, 1981, Huber 1985) and transformations of additive models (Stone 1985, Tibshirani 1988). New composition schemes are suggested which combine the positive benefits of the above methods.

Although there are interesting analogies of statistically estimated network functions with the activity of networks of living neurons, we shall not constrain our network functions to be biologically viable models. Instead the focus is on the development of empirical modeling capabilities for network function so as to represent the input/output behavior of a wide range of complex systems for scientific and engineering applications.

Mathematical limitations of high-dimensional estimation are discussed. Bounds from nonparametric statistical theory show that reasonably accurate estimation uniform for all smooth functions (e.g. functions with bounded first partial derivatives) is not possible in high dimensions with practical sample sizes. Network strategies avoid some of the pitfalls of high-dimensionality by searching for structures parameterized by lower dimensional forms. The advantage is that for high-dimensional problems the variance (estimation error) associated with such networks can be much smaller than associated with more traditional approaches. As for the bias (approximation error), the evidence is that for many practically occurring functions accurate network approximations exist, in spite of the theoretical fact that high-dimensional functions can possess sufficiently irregular structure so as to preclude accurate estimation.

Some dynamic network models (such as the Hopfield network 1981) are differential equations (or difference equations) resulting from cycles present in the interconnected network. In this paper we restrict attention to static network models which have no loops in the network. Thus the network is a tree of interconnected functions which implements a single input/output function, which may be adjusted by the empirical estimation process, but otherwise is static.

2. Block Diagrams

We present a hypothetical network to get oriented to some terminology and notation. A function which is defined as a *composition*, such as

$$f(x_1, x_2, x_3, x_4) = g_0(g_1(g_3(x_1, x_2), g_4(x_1, x_3, x_4)), g_2(g_4(x_1, x_3, x_4), g_5(x_4))),$$

may also be written in terms of *intermediate variables*

$$f = g_0(z_1, z_2)$$

$$z_1 = g_1(z_3, z_4),\ z_2 = g_2(z_4, z_5)$$

$$z_3 = g_3(x_1, x_2),\ z_4 = g_4(x_1, x_3, x_4),\ z_5 = g_5(x_4),$$

or it may be drawn as a network diagram (Fig.1):

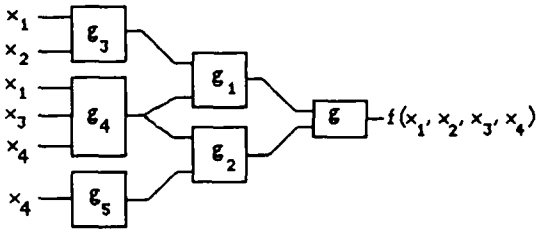


Fig.1. Example Network

The *layers* of a network are the sets of functions which occupy the same depth in the tree.

For a general notation for network functions, in which the indices on a basic function specify the position of the function in the tree relative to the root node, see Lorentz (1966). He called network functions *superposition schemes*. Lorentz made fundamental contributions to the theory of representing functions by compositions which are discussed later in this paper.

Representations for network functions are not unique. For instance, if some of the basic functions are absorbed into the functions to which they are input, then fewer elements are obtained, but the new elements have possibly greater input dimension.

Motivated by the application to modeling human vision, Rosenblatt (1962, ch. 4) called networks with arbitrary

[1]Work supported in part by an Office of Naval Research grant N00014-86-K-0670 and by a National Science Foundation Postdoctoral Research Fellowship.

elemental functions *perceptrons* (although subsequently the term has been used to refer to just one type of network with thresholded linear elements that Rosenblatt extensively studied). Our definition differs slightly from Rosenblatt's in that he allowed transformations to occur on the branches (interconnections) of the network. Such networks are represented in our form either by defining additional single input nodes or by absorbing each such transformation into the node to which the branch is directed.

3. The Building Blocks

For learning networks it is important to choose elements of the network with sufficiently general form that the resulting networks can approximate nearly any function of interest. It is also important to choose these elements with sufficiently small dimension or complexity that they can be accurately estimated. Different approaches to resolving the tension between these two seemingly conflicting objectives result in a variety of different learning network schemes.

Let the function $g(z)$ denote an element of the network, where z is the vector of intermediate variables (outputs from preceding elements or sometimes original input variables) which are input to the given node. The most common forms of elements roughly can be categorized as parametric or nonparametric.

Parametric elements: These are basic functions $g(z, \theta)$ which depend on a vector of unknown parameters. The parametric elements which have been proposed for learning networks usually take one of the following forms:

$$g(z, \theta) = h\,(\Sigma\, \theta_k z_k + \theta_0) \qquad (1)$$

$$g(z, \theta) = \Sigma\, \theta_k \varphi_k\,(z) \qquad (2)$$

or, more generally,

$$g(z, \theta) = h(\Sigma\, \theta_k \varphi_k(z)) \qquad (3)$$

where φ_k, $k = 1,\ldots, m$, and h are fixed functions. The two most common choices for the φ_k are linear terms (coordinate functions), so that the sum simply implements a linear combination of the inputs as in (1), or polynomial terms of moderate degree. The nonlinear function h is typically chosen to be a nondecreasing function bounded by one (such as a unit step function) -- this is frequently incorporated in networks intended for binary classification. The parameters of each element are estimated from observed data, typically by a least squares or likelihood based criterion. The specific method used to estimate the parameters depends on the probabilistic structure of the data, the network synthesis strategy, and the intended use of the network (see section 4 below).

Nonparametric elements: Some of the element functions $g(z)$ may be regarded as unknown and constrained only in terms of basic smoothness properties (e.g. bounded derivative), or in some cases g is modeled as a stochastic process indexed by z (a Bayes formulation). Such functions are estimated by a smoothing technique such as local linear fits, smoothing splines, variable kernel estimation, truncated trigonometric series, variable degree polynomials, or stochastic process estimation. Typically parameters of the smoothing technique are selected by a criterion such as cross-validation, predicted squared error, or penalized likelihood. With nonparametric elements it is important that the dimension of the z variables be kept to a minimum. (Otherwise the statistical theory indicates that it would be difficult to estimate these element functions.)

Mixed parametric/nonparametric: In this case both types of elements appear in the network. A particularly interesting approach is to combine nonparametric elements, each of which depends only on one variable, with elements which implement linear combinations of many variables. It will be seen that networks of this mixed structure have the potential to approximate any function.

We use the notation $f(\underline{x}, \theta)$ to refer to the complete network function where $\underline{x}$ is the vector of all original input variables and θ is the vector of all parameters which appear in the network.

4. The Structure of the Data and Objective of Network Estimation

In practice, networks are estimated from a training sample of observations of relevant variables. The sample is tpically a sequence of input/output pairs $(\underline{X}_1, Y_1), \ldots, (\underline{X}_n, Y_n)$ where each $\underline{X}$ is a d-dimensional vector. We focus on the case in which the observations are independent, each with the same probability distribution $P_{\underline{X},Y}$. (Certain problems involving data with stationary serial dependencies can also be treated, in which case the relevant distribution is the conditional distribution given the past.) This probability distribution is assumed to depend on an unknown function $f(\underline{x})$: it is this function which neural networks seek to approximate. The assumed nature of this function depends on the objective of the problem (e.g. regression, prediction, classification, density estimation) and the criterion by which performance is measured.

Perhaps the most common use of learning networks is to seek a function $f(\underline{x})$ to minimize the *mean squared error* $E(Y - f(\underline{X}))^2$: that is, the function we wish to estimate is the conditional mean $f(\underline{x}) = E\,[Y \mid \underline{X} = \underline{x}]$. For problems of *curve fitting, regression,* or *prediction* this conditional mean function has traditionally been the principle object of interest for learning networks. (For certain *time-series* prediction problems the desired function takes on the specific form $f(\underline{x}) = E[Y_t \mid Y_{t-1} = x_1, \ldots, Y_{t-d} = x_d]$). In particular, this framework (associated with a squared error measure of loss) is appropriate when a function $f(x)$ is measured subject to (mean zero) Gaussian error at randomly distributed design points.

For *classification* problems, an optimal discriminant function is one for which the overall *probability of error* is minimized. Most often, learning networks have been utilized to seek an indirect solution to the classification problem by using the mean squared error as the criterion. For two-class classification with $Y \in \{0,1\}$ the conditional mean function reduces to the optimal discriminant $f(\underline{x}) = P\,[\,Y = 1 \mid \underline{X} = \underline{x}]$. Nevertheless, it may be more appropriate to seek to estimate the logistic regression function $f(\underline{x}) = log(P\,[Y = 1 \mid \underline{x}]/(1 - P\,[Y = 1 \mid \underline{x}]))$ using likelihood-based criteria. In principle, *probability density estimation* can also be handled using learning networks and a likelihood criterion, in which case f is taken to be the logarithm of the joint density function of the random vector.

The intended use of estimated network functions $\hat{f}$ may dictate probability models and performance objectives other than those indicated above. For instance the object may be to *search* for the extreme points of a function f by using the extreme points of $\hat{f}$. For problems in *vehicle guidance,* the function $\hat{f}$ might estimate parameters of an optimum (two-point boundary-value) guidance law as a function of current and desired final vehicle states (in situations where the optimum f can only be obtained by extensive off-line iteration), in which case the ultimate performance objective is to minimize the final miss distance, rather than to minimize the mean squared error of the parameter estimates. Nevertheless, learning network methodologies have proven successful in some of these contexts (see R. L. Barron and Abbott 1988).

Most network algorithms have been designed for regression or classification with minimum mean squared error as the performance objective, and our attention will be focused primarily on this case.

5. Criteria for Network Estimation and Selection

Here we discuss model selection criteria needed for the estimation of network functions. Without the use of an appropriately penalized performance criterion, an overly complex network may be estimated which accurately fits the training data but will not prove to be accurate on new data.

Predicted squared error: If a network structure $f(x, \theta)$ is fixed and if the total number of parameters k is small compared to the sample size n, then the minimum mean squared error $min_\theta E\,(Y - f(X, \theta))^2$ is approximately achieved by seeking parameter estimates $\hat{\theta}$ that produce the minimum average squared error on the training set, $TSE = \frac{1}{n}\sum_{i=1}^{n} (Y_i - f(X_i, \hat{\theta}))^2$. However, if k is large compared to n, then the model may have small error on the given data, but it is likely to have large error on future data from the same distribution. This phenomenon is partly explained by noting that, under certain conditions (namely that the network depends linearly on the parameters and the true function $f(x)$ happens to be a member of the given k-dimensional family with error variance $\sigma^2 = E\,(Y - f(X))^2$), the mean squared error of an estimated network of fixed dimension k is not equal to the error variance σ^2 but rather is equal to $E\,(Y - f(X, \hat{\theta}))^2 = \sigma^2 + (k/n)\sigma^2$: see Mallows (1973), A.R. Barron (1984). This leads, in view of the fact that under the same conditions $E\,(TSE) = \sigma^2 - (k/n)\sigma^2$, to the predicted squared error PSE criterion as an unbiased estimator of the future performance:

$$PSE = TSE + \frac{2k}{n}\sigma^2. \qquad (4)$$

This criterion is very similar to (and in some cases equivalent to) the C_p statistic proposed by Mallows (1973), the generalized cross-validation criterion of Craven and Wahba (1979), the final prediction error of Akaike (1970), and a specialization of the AIC proposed by Akaike (1973). For a recent treatment of these various criteria with emphasis on generalized cross-validation see Eubanks (1988, ch. 2). Calculations similar to those in Akaike (1973) show that PSE continues to be an asymptotically unbiased estimator of the mean squared error $E\,(Y - f(X, \hat{\theta}))^2$ even if $f(x, \theta)$ is not a linear function of θ, provided this function is sufficiently smooth.

Unfortunately, if the network function is selected so as to minimize PSE among a collection of functions of various parameter dimensions, then there is no general guarantee that the resulting minimum PSE will be an accurate estimate of the mean squared error of the estimated function. Indeed, if the true function f is a member of one of the finite-dimensional network families, then the PSE criterion has a tendency to overestimate the dimension (see Atkinson 1980, 1981). On the other hand, the work by Shibata (1984, 1986) shows in related contexts that if the true function $f(x)$ is not exactly representable by any of the finite dimensional models in a sequence $f_k(x, \theta_k)$ *for* $k=1,2,\ldots$ (but can nevertheless be approximated by such models), then selection of $\hat{k}$ by a criterion of the form given above is *optimal* in the sense that the resulting expected squared error $E\,(f(X) - \hat{f}(X))^2$ is asymptotically equivalent to $min_k E\,(f(X) - f_k(X, \hat{\theta}_k))^2$ as $n \to \infty$. It is not known if the results of Shibata carry over to the estimation of network functions. Nevertheless, in our experience with numerous practical cases (see Barron et al. 1984), networks selected by minimizing PSE have approximately minimal average squared error on independent sets of test data (in the sense that if the growth of adaptively synthesized networks is halted on an earlier layer or allowed to extend to a larger number of layers, then a significant increase in the average squared error on the test set does not usually occur).

If the error variance σ^2 is not known, an estimate $\hat{\sigma}^2$ can be used in its place in the PSE criterion; however, to avoid overfit care must be taken to avoid having $\hat{\sigma}^2$ much less than σ^2; in particular, $\hat{\sigma}^2$ should not be varied during the process of selecting k (A. R. Barron 1984). We suggest that nearest neighbor regression be used prior to network synthesis to determine a rough estimate of the error variance with the desired properties. To permit consistent estimation of f in the case that it can be exactly represented by a finite dimensional network (as well as in the case that it can be arbitrarily well approximated by networks of sufficient dimensionality) other criteria should be used which place a greater penalty on the dimensionality of the model (e.g. $\frac{k}{n} log\ n$ instead of $\frac{2k}{n}$). Criteria significantly different from PSE will not possess the optimum rate property of Shibata in the context that he considers; however, it is not known to what extent the convergence rate is slowed.

Likelihood based criteria: Suppose the random vectors $(\underline{X}_i, Y_i)$ have a conditional probability density function $p(y \mid \underline{x}, f)$ which depends in a known way on the value of f (whereas the true function $f(\underline{x})$ may be unknown). Let $f(\underline{x}, \theta)$ be a given network structure with a k-dimensional parameter θ. Assume that $\hat{\theta}$ is estimated so as to maximize the likelihood $p(Y^n \mid \underline{X}^n, f(\cdot, \theta)) = \prod_{i=1}^{n} p(Y_i \mid X_i, f(X_i, \theta))$. Define the Akaike information criterion (Akaike 1973) by

$$AIC = -\ log\ p(Y^n \mid \underline{X}^n, f(\cdot, \hat{\theta})) + k \qquad (5)$$

and define the minimum description length criterion (Rissanen 1978, 1983) by

$$MDL = -\ log\ p(Y^n \mid \underline{X}^n, f(\cdot, \hat{\theta})) + \frac{k}{2} log\ n. \qquad (6)$$

These criteria are used to choose between models of various dimensions. Akaike derived the AIC as an asymptotic bias correction for the estimation of expected entropy loss, in much the same manner that PSE is an asymptotic bias correction for the estimation of expected squared error. Rissanen derived the MDL criterion as the length of a uniquely decodable code for quantizations of the data Y^n given the data X^n (ignoring terms which are asymptotically constant for k bounded). Unlike the optional Shannon code, Rissanen's code does not require knowledge of the function f. Instead, the MDL code uses quantized maximum likelihood estimates of the parameters of the function as a preamble of the code (using $\frac{1}{2}$log n bits per parameter). The criterion can also be derived as an asymptotic approximation for the Bayesian test statistics which minimize average probability of error in the selection of the model (see Schwarz, 1978, Clarke and A.R. Barron, 1988).

The validity of the derivations of AIC, MDL, and Bayes criteria require smoothness conditions. In particular the sample Fisher information matrix $\hat{I}$ of second partial derivatives with respect to θ of $-\frac{1}{n} log\ p(Y^n \mid \underline{X}^n, f(\cdot, \theta))$ (evaluated at $\theta = \hat{\theta}$) should be positive definite. A more precise form of the MDL or Bayes criterion uses $\frac{1}{2} log\ det(\hat{I})$ instead of $\frac{k}{2} log\ n$.

For regression with a Gaussian error distribution and known error variance, the AIC reduces to the PSE criterion and MDL reduces to a criterion equivalent to

$$TSE + (\tfrac{k}{n}\log n)\sigma^2. \qquad (7)$$

For classification problems with $Y \in \{0,1\}$, likelihood based criteria are defined by using the Bernoulli model $p(y \mid \underline{x}, f) = (f(\underline{x}))^y(1 - f(\underline{x}))^{1-y}$ (in which case care must be taken to use networks with $0 < f(\underline{x}) < 1$). The equally general logistic model $p(y \mid \underline{x} f) = e^{yf(\underline{x})}/(1 + e^{f(\underline{x})})$ may be preferred for classification problems, since it forces satisfaction of the probability constraints $0 < p < 1$ without constraining the function f. For logistic regression the minus log-likelihood takes the form $\sum \log(1 + e^{f(\underline{X}_i, \theta)}) - \sum Y_i f(\underline{X}_i, \theta)$, which is minimized (e.g. by Newton's method in the context of various synthesis strategies) and then penalized by k or $\frac{1}{2}\log n$ as appropriate for the desired criterion.

Complexity regularization: In A.R. Barron (1985) the minimum description length criterion is extended to nonparametric contexts in which the description length need not reduce to the form of (6). Consistency results are obtained in A.R. Barron (1985, 1987) which show convergence ($as\ n \to \infty$) of distributions estimated by the complexity regularization. The specialization of the convergence results to the case of estimation of network functions is given in the Appendix.

6. Main Strategies for Network Synthesis

There are two main strategies for the synthesis of networks depending on whether the structure of the network is fixed or allowed to evolve during the synthesis process.

Fixed networks: In this approach a fixed composition structure (often relatively large) is preselected with the hope that the desired function can be accurately approximated by networks of the selected form. The problem of choosing parameters of the network so as to optimize a performance criterion may be regarded as a *global search of a highly multimodal surface.* In general, global convergence is difficult to guarantee; nevertheless, by choosing a network function which depends smoothly on the parameters it is often feasible to estimate sufficiently accurate network functions by certain global search techniques (e.g. techniques which alternate global random and local gradient search). Other methods for estimating network functions attempt to localize the search within each unit of the network by defining target values for each elemental function. More specifics are given in section 7 below.

The advantage of the fixed network approach is that certain structures are known to have the ability to approximate any continuous function (see section 13). However, for moderate sample sizes, these fixed structures may have too large a parameter dimension for the least squares or maximum likelihood estimators to be accurate. In this case, to prevent irregularity of the estimated function, it is useful to constrain the parameters so that the resulting network function is smooth or to penalize the performance criterion by incorporating a term for the lack of smoothness (e.g. the sums of squares of first partial derivatives of the network functions at the observations). Of course the criteria mentioned in section 5 above are not adequate when the dimension of the network is fixed in advance.

Adaptive networks: In this approach, the attempt is to estimate networks of the right size with a structure evolved during the estimation process to provide a parsimonious model for the particular desired function. Typically, the network is estimated one layer at a time, with the elements on each given layer selected to minimize the predicted squared error or complexity regularization criterion. The basic idea is that once the elements on a lower level are estimated, and the corresponding intermediate outputs z are computed, then the parameters in a given element $g(z, \theta)$ may be estimated by usual least squares or likelihood maximization techniques. It is most common for the elements on each layer to be greedily trained to attempt to best estimate the desired final output, even though the outputs of these elements are combined on succeeding layers. On the other hand, some methods developed in statistics select the element functions so as to work best in linear combination with the previously selected elements on a given layer.

Practical experience shows clear advantages of the adaptively synthesized networks over some of the globally optimized fixed network structures. (However, certain theoretically appropriate fixed structures have yet to be tried in practice; also, the smoothness penalty criteria have yet to be utilized with the larger fixed networks.) In most instances the adaptively synthesized networks are more parsimonious. Parts of the network which are inappropriate or extraneous for statistically modeling the given data are automatically not included in the final network. The drawback of the adaptive strategies is that they cannot be guaranteed to work. It is possible to find counterexamples of data corresponding to functions which are exactly modeled by a two-layer network, but no non-trivial first layer elements are selected by a given adaptive synthesis strategy.

Mixed adaptive/global strategies: After the best elements on each layer are computed, a numeric search can be used to update the estimates of parameters for ancestral nodes on earlier layers. An iterative scheme that alternates between estimation of the parameters of the given element and the estimation of the parameters of the ancestral nodes is suggested by the projection pursuit algorithm and its generalizations (see sections 10 and 12).

7. Some Early Network Developments

While linear models for regression and thresholded linear models for classification (e.g. of the form (1), (2), or (3)) have been long used in statistical practice (with the beginnings of the modern understanding due in large part to R.A. Fisher (1922, 1934, 1936) who introduced measures of statistical efficiency, explained the efficiency of maximum likelihood estimation, and derived the linear discriminant function for multivariate Gaussian classification), these same linear models were reintroduced (unfortunately with comparatively inefficient estimators) in the 1950's and 1960's as a basic ingredient in learning network models. The new and interesting twist was that more general classes of functions were modeled by combining these simpler models into a network. Here we mention some of the development which occurred in this period.

The forerunners in the network modeling field were McCulloch and Pitts (1943), who introduced the thresholded linear function as a model for the behavior of a neuron and, in that paper, analyzed the model not so much for its biological viability, which was discussed only briefly, but rather (in the language of theoretical computer science) as a basic computational unit with the property that any predicate with finite domain could be implemented by a network of such units.

There was a surge of interest in methods for the inference of networks (Hebb 1949, Ashby 1952, Farley and Clark 1954, Minsky 1954, von Neumann 1956, Rosenblatt 1957, Lee and Gilstrap 1960) culminating in some interesting and successful multiple layer estimation methods in the early 1960's due to Rosenblatt (1962), Widrow et al. (1960, 1962, see also 1987), and R.L. Barron et al. (1964, see Moddes et al. 1965, Gilstrap 1971, Barron et al. 1984). Although some of the networks due to Rosenblatt and Barron et al. used more general elemental functions than the original thresholded linear function, they did share the form (3) (transformed variables were combined linearly using free parameters). These heuristic multi-layer

methods were not well understood theoretically and (with the exception of Rosenblatt's book) they were not widely disseminated at that time. We emphasize that contrary to the popularly held current belief (initiated in the book by Minsky and Papert 1969 and perpetuated by statements as in Rumelhart et al. 1986, p.321), powerful rules were found for the estimation of multiple layer networks.

The methods of Widrow et al. and Rosenblatt for binary classification possessed many similarities. In particular, both authors exclusively utilized recursive estimation strategies in which the parameter estimates are updated with each new observation by an error correction procedure analogous to the Robbins-Monroe (1951) stochastic approximation (but without the full statistical efficiency known to hold for recursive least squares or recursive implementations of maximum likelihood). Moreover, both approaches were amenable to clear theoretical proofs of convergence properties in the case of single element networks (these results are well-explained in Nilsson (1965) and Duda and Hart (1973)). Widrow used a stochastic gradient method which he called the least mean squares (LMS) algorithm. Rosenblatt used a method (related to relaxation procedures for solving linear inequalities, Agmon 1954), which he called the perceptron algorithm: it finds a hyperplane which perfectly separates the two classes whenever the classes are linearly separable. The non-convergent behavior in the non-separable case was analyzed by Efron (1964).

For multiple layer networks the method of Widrow et al. (1960, 1962) was only explained in the case that first layer elements are adjustable and the succeeding layers are preselected. Widrow used iterations of his strategy to handle also the more general estimation problem, but this approach was not published until Widrow 1987, to which we refer the reader for a description.

For two and three layer networks of thresholded linear elements, Rosenblatt (1962, ch. 13) developed an algorithm which he called *back-propagating error correction* (unfortunately, this name recently has been reused for another algorithm for network estimation, as mentioned below). The objective of his method is recursively to estimate desired outputs for every element as well as to estimate the parameters. Naturally, given a desired output of an element Rosenblatt updates the parameter estimates in the element by his perceptron algorithm (here a parameter update occurs only if the actual output differs from the desired output). On the other hand, if the output of an element does match the desired value, then depending on whether the resulting final output of the network is in error, the desired intermediate variable is adjusted to reduce this error (again as in the perceptron algorithm but with the role of parameters and variables reversed). (Randomization is used to avoid certain degeneracies. In particular, with each step no update action is taken with probability $0<p<1$.) Rosenblatt advocated cycling through the data and the elements of the network in such a way that each combination (of datum and network element) potentially would be considered infinitely often. He presented a theorem (Rosenblatt, p. 294) to the effect that if the data are separable by the network (i.e. there exist parameter values for which the network function correctly classifies every point), then his estimation strategy will find such an error-free solution in a finite number of steps (with probability one).

The approach developed by R.L. Barron et al. (1964) and further explained in Moddes et al. (1965), Gilstrap (1971), and Barron et al. (1984) solved the multilayer network estimation problem by global search to minimize the sum of squared errors $\Sigma(Y_i - f(X_i, \theta))^2$. Barron et al. introduced an algorithm called *guided accelerated random search* (GARS) which alternated between global random search (using a spherical normal distribution centered at the current best point) and local gradient search (for which convergence was accelerated by a halving/doubling algorithm for the step size and by adjusting a variable subset of the parameters at the different steps). The particular elemental functions originally used by R.L. Barron et al. were quadratic functions in two variables $g(z,\theta) = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \theta_3 z_1 z_2$. A spirally-connected network with 24 input variables and seven layers was constructed (see fig. 2). Using 25-50 observations of simulated reentry vehicle positions during a given time frame $(t, t-\Delta t,..., t-7\Delta t)$, networks were constructed to predict the final position and impact time of the vehicle. The parameters of the networks were constrained to values in the interval between -1 and +1. The GARS search routine converged to essentially the same extremum of performance for each of many randomly selected initial parameter vectors, suggesting that a non-unique global optimum was reached. Performance on an independent test set of observations suggested that despite the complexity of the network, and the small sample size, the estimated function was not overfit to training data. (However, overfit problems were later experienced with these large fixed networks on some industrial process modeling problems -- these experiences led in the early 1970s to the adoption of adaptive synthesis strategies discussed below.)

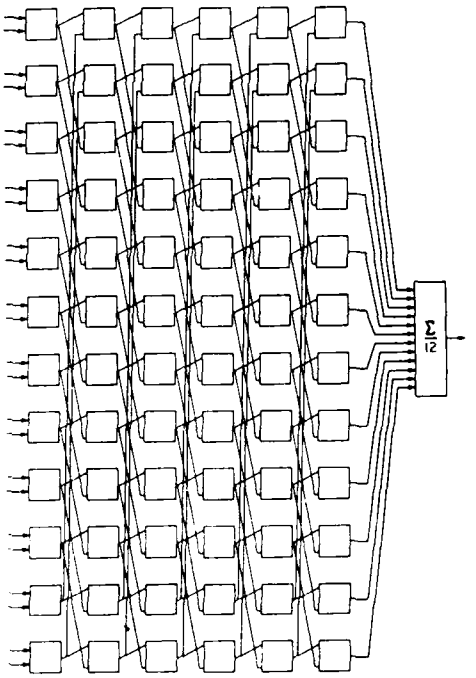


Fig. 2. Uniform Spiral 72-Element Network

The network of fig. 2, which consists of quadratic two-input elements, represents a family of sixth-degree polynomials. Since the network contains a total of 288 parameters, this family is a relatively low-dimensional manifold in the complete (593,775 dimensional!) family of sixth-degree polynomials in 24 variables. Nevertheless, the network had more than enough flexibility to yield accurate approximations for the specific application to re-entry vehicle trajectory predictions.

8. The Current Fashion

In recent work Rumelhart, Hinton, and Williams (in Rumelhart et al. 1986, ch. 8) propose that an implementation of the gradient descent algorithm be used to attempt to

minimize the sum of squared error for multiple layer feedforward networks. They use element functions of the form (1) with h equal to a logistic function: this choice is viewed as a smoothing of the step function to obtain a differentiable function of the parameters. Since the network is a composition of functions, the derivatives required for the gradient method are determined by the chain rule of calculus (starting at the final node and propagating back to the parameters in the first layer). Although it is recognized that the gradient method may be inappropriate in general for highly multi-modal surfaces, Rumelhart et al. found that it worked adequately on the simple examples that they considered. Hinton and Sejnowski (in Rumelhart et al. 1986, ch. 7) propose that a sequential random search algorithm (simulated annealing) be used to estimate the parameters of a Hopfield style network; they call their learning network a Boltzmann machine. These papers (see Rumelhart et al. p. 32) give the impression that multilayer search strategies for networks are novel to the 1980s. Clearly this is false in view of the methods we have discussed. In our experience (beginning in the 1960s) a combination of random and derivative-based search strategies, as in the GARS algorithm, is an effective technique for globally optimizing networks. In any event, much of the recent work (as in Rumelhart et al.) has ignored the developments in the 1970s and 1980s of the adaptive network strategies and the nonparametric statistical methodologies for specific network structures.

9. Networks with Adaptively Synthesized Structure

With the propensity of large fixed networks to result in overfit estimates, attention was turned in the 1970s to networks for which the structure is adaptively determined from the data. Such network strategies were introduced by Ivakhnenko (1971) and their development in the U.S. is traced in Barron et al. (1974, 1975, 1984, 1987).

The elements extensively utilized in these adaptively synthesized networks are second- and third-order polynomial functions in two variables. (One and three variable elements are also used in recent implementations.) For the method to work, the number of inputs of each element must be restricted so as to avoid a combinatorial explosion in the number of possibilities that the algorithm must check.

In brief, the basic strategy (using elements involving two variables) is depicted in fig. 3. On the first layer, all possible pairs of the inputs are considered and the best k_1 are temporarily saved. On the succeeding layers, all possible pairs of the intermediate variables z from the preceeding layer(s) are considered and the best k_2 (k_3, etc.) are saved. Finally, when additional layers provide no more improvement, the network synthesis stops. The final network consists only of the ancestors of the final element.

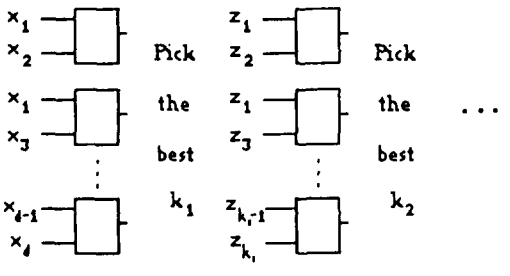


Fig. 3. An Adaptive Network Synthesis Strategy

In the original Ivakhnenko algorithm, the parameters within each element were estimated so as to minimize on a training set of observations the sum of squared errors of the fit of the element to the final desired output. Cross-validation on a separate testing set was used to rank and select the best elements on each layer and to select the number of layers. (Ivakhnenko called this division of the data into sets with different purposes in network estimation the *group method of data handling*, GMDH.) The need to construct complete quadratic polynomials for every pair of variables forced early implementations of the algorithm to restrict the number k of temporarily saved intermediate variables to be typically not more than 16.

Later algorithms developed by A.R. Barron (1979-1982, Polynomial Network Training Routine, PNETTR III and IV, Adaptronics, Inc.) incorporated a predicted squared error PSE criterion (related to the criteria of Akaike and Mallows as discussed above) at every phase of element selection in the network. Moreover, a method was developed whereby candidate pairs are prescreened before each layer (according to their predicted error in linear combination) thereby permitting more elements to be considered on each layer (typically k is between 30 and 60). This also permitted more complicated element calculations, i.e. third-degree polynomials with subset selection by the PSE criterion. Also the saved elements from all preceeding layers are candidate inputs to a given layer. Moreover, some one- and three-input elements are considered on each layer. The PNETTR algorithm was extensively applied to problems in nondestructive evaluation of materials, modeling of material characteristics, flight guidance and control, target recognition, intrusion detection systems, and scene classification; see Barron et al. (1984) and the references cited there. For an application of an earlier version of the algorithm to weather forecasting see A.R. Barron et al. (1977).

The more recently developed algorithm by J.F. Elder IV (1985-present, Algorithm for Synthesis of Polynomial Networks, ASPN, Barron Associates, Inc.) permits a choice of a minimum complexity or predicted squared error criterion. This algorithm has more user flexibility in the choice of one-, two-, or three-input elements and in the form of the polynomial elements (e.g. the degree may be adjusted within certain limits). Moreover, at each layer a new element is considered which is a linear combination of all elements on the preceeding layer.

Currently, a major applications thrust is use of adaptively-synthesized polynomial networks to initialize and/or re-initialize (in real time) two-point boundary-value guidance solutions for flight vehicles (R.L. Barron and Abbott 1988). Polynomial networks are trained off-line on a library of simulated optimum trajectories and interrogated on-line with information about existing and desired vehicle states. Interrogation yields numerical values of six initializing adjoint variables (Lagrange multipliers) in a calculus of variations formulation of the trajectory optimization solution. Because each new interrogation answers the *optimum-path-to-go* question, a guided trajectory need not be restored, when disturbed, to a preconceived nominal path, and optimality of trajectory energy management and accuracy of guidance are not compromised by disturbances within maneuvering limits of the vehicle. In the two-point-boundary-value guidance application, the role of the polynomial network is to compress a large library of multivariate trajectory information and render it in a form (the network) suitable for virtually instantaneous look-up and interpolation.

Fig. 4 is a diagram for networks trained to estimate two of the initializing adjoint variables for a specific flight vehicle guidance application. These networks were synthesized from a data base of 435 observations of the candidate variables. Ten variables were selected by ASPN for inclusion in the final model. The information presented in each box refers respectively to the index of the element (in the list of elements saved by ASPN during synthesis), the type of element (in terms of number of inputs), and the number of terms in each cubic expression after pruning according to a PSE

criterion. The "white" element computes a linear combination of its inputs.

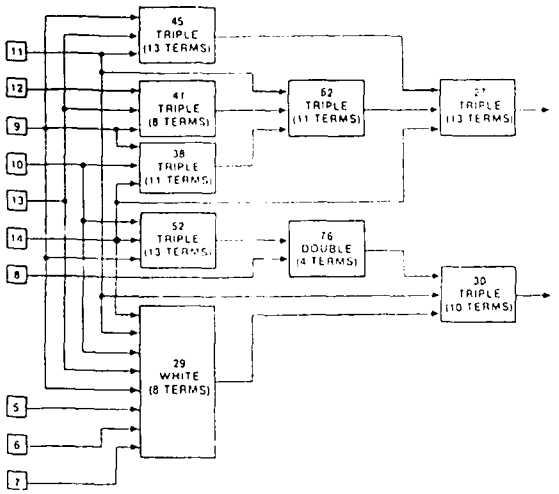


Fig. 4 An Adaptively Synthesized Polynomial Network

10. Projection Pursuit

The projection pursuit algorithm of Friedman et al. (1974,1981,1984) which is so popular in statistical circles has not previously been discussed in the context of learning networks. This algorithm adaptively synthesizes a three-layer network in the form of fig.5. The first-layer functions implement linear combinations $\Sigma\theta_{jk}x_j$ for ordinary projection pursuit *(or $\Sigma\theta_{jk}\phi_{jk}(\underline{x})$ for a generalization of projection pursuit* to be discussed below). The second-layer functions $g_k(z)$ are nonparametrically estimated functions of one variable. Finally, the third layer simply takes a linear combination $\Sigma\beta_k g_k$. Thus the function implemented is $f(\underline{x}, \theta, \beta) = \sum_k \beta_k g_k(\sum_j \theta_{jk} x_k)$.

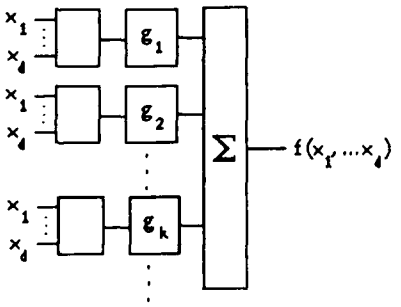


Fig. 5. Network Diagram for Projection Pursuit

The estimation strategy of projection pursuit proceeds vertically through the levels indicated in fig.5. On each level, an iterative Gauss-Newton algorithm is employed which alternates between estimation of the parameters θ from the first layer and the function g_k from the second layer so that in linear combination with the preceeding levels the fit is optimized (using the sum of squared errors or a likelihood criterion). Here the use of the optimized linear combination $\Sigma\beta_k g_k$ is a relaxation method suggested by Lee Jones (1986) as an improvement over the original method (which estimates g_k to fit the error $y - (g_1 + \ldots + g_{k-1})$).

To estimate the functions $g(z)$, Friedman et al. utilize a nonparametric smoothing technique involving locally linear functions (the linear fit at an arbitrary point z is estimated using the data in a neighborhood of that point). Nevertheless, the methodology also works with other one-dimensional nonparametric estimation techniques such as smoothing splines or variable degree polynomials.

Projection pursuit provides an excellent example of a learning network with both parametrically and nonparametrically estimated elements. Also, it demonstrates an effective iterative strategy for estimating the elements of a layer of a network to work well in combination with each other rather than in isolation.

An advantage of projection pursuit networks is that they have been amenable to theoretical examination of some of their approximation properties (Huber 1985, Donoho and Johnstone 1985, Jones 1987), although much work remains to be done in this direction. In particular it is known that any square integrable function can be approximated by a theoretical analog of projection pursuit, provided sufficiently many (vertical) levels of the network are utilized; however, the analogous result for data-driven estimation has yet to be established.

11. Additive Models and Transformations

Additive models represent functions of the form $\Sigma g_k(x_k)$, where in general the one-dimensional functions g_k are unconstrained and in practice usually are estimated non-parametrically. (In contrast, linear models estimate only the coefficients of linear combinations of fixed functions.) The theory for the estimation of additive functions is developed in Stone (1985). In particular, Stone demonstrates the surprising result that, unlike general functions of d variables, additive functions can be estimated with a convergence rate for the expected squared error which is as good as the rate which can be obtained for the estimation of one-dimensional functions ($n^{-2r/(2r+1)}$ instead of $n^{-2r/(2r+d)}$ where n is the sample size, r is the assumed order of smoothness, and d is the dimension; see section 14 below). Moreover, Stone showed that although not every function is additive, a best additive approximation to a function exists and can be estimated at the indicated rate. Stone's approach to estimating the additive functions is to use finite dimensional linear spaces of functions (such as splines, polynomials, or truncated trigonometric series – in particular Stone uses splines), so that the resulting additive approximation is then written in terms of a linear function of many fixed basis functions, in which case traditional least squares projection becomes applicable.

Winsberg and Ramsay (1980) and Tibshirani (1988) generalize additive approximation by permitting monotone transformations $h(y)$ of the dependent variable. By inverting this transformation, an approximation to the dependent variable is obtained in the form depicted in fig. 6 with $g=h^{-1}$. A related model is in Breiman and Friedman (1985) where noninvertible transformations h are permitted.

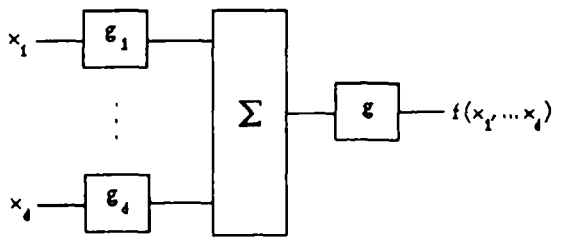


Fig. 6. Network for Transformations of Additive Models

Networks as in fig. 6 can be estimated by alternating between estimates of the transformation g and the first layer

functions g_k using methods similar to projection pursuit. In particular, suppose finite series approximations are used for each of the functions g_k. Given a current estimate of g (which is assumed to be a differentiable function), a Gauss-Newton type algorithm can be used for the estimation of the coefficients in a finite series approximation of the g_k. Then, given the current g_k, the new estimate of g can be obtained by any of several nonparametric methods (e.g. least squares projection onto a linear space of approximating functions, local linear smoothing, etc.). These steps are then iterated until only negligible improvement in the optimization criterion is observed.

Our purpose for mentioning additive models in the context of networks is that this structure is the one which is best understood theoretically (except perhaps for linear discriminate functions and linear regressions which have even less approximation capabilities) and, moreover, the additive structure is a basic building block for more elaborate networks which show some promise. Although additive models cannot represent interactions between variables, interactions can be obtained by taking sums of transformations of additive models as seen below.

12. Generalizations

It appears to us that certain extensions to the network forms of projection pursuit or transformations of additive functions lead naturally to a particular network structure which is known to have powerful approximation capabilities. The statistical estimation strategies associated with projection pursuit and additive models then lead to estimation strategies for these more complex network forms.

In particular, consider networks of the form given in fig. 7. This form may be regarded as a projection pursuit network, generalized to allow transformations of the original variables on the first layer. Using series approximations (e.g. polynomials) for these transformations, the projection pursuit estimation algorithm becomes applicable to this network as discussed in section 10. Alternatively, the network of fig. 7 may be thought of as a composition of additive functions. Specifically, the network consists of $2d+1$ additive functions with outputs $z_1, z_2, \ldots, z_{2d+1}$, say, which become the inputs to a final additive function with output f. Whereas none of the lower layer additive portions of the network can approximate every function, the composition of these functions can approximate any continuous function as discussed in section 13 below. In principle, any of the methods for estimating transformations of additive models can be used to estimate the k'th such function by fitting the model to the error resulting from the sum of the previous k-1 models. However, such iterative approximations may require more than the $2d+1$ levels indicated by the theory.

A specific implementation of a generalized projection pursuit algorithm which incorporates some of the features mentioned above is being developed by A.R. Barron and Gayle Nygaard. It will permit the use of polynomial, spline, or trigonometric series approximations for any of the transformations of the network. A new feature of this algorithm is that, when estimating g_k in fig. 7, the transformations $g_1, g_2, \ldots, g_{k-1}$ are backfitted to provide the best additive combination by projecting to sums of basis functions in the manner of Stone (1985). Moreover, after each transformation is estimated, a backward stepwise rule (using a penalized squared error or complexity criterion) is used to prune unnecessary terms from each element. In view of the relatively large (but fixed) size of the network structure, this pruning of the number of coefficients is essential to avoid overfit with moderate sample sizes. The most important generalization is to permit nonparametrically estimated transformations of the variables so as to achieve "projections" to surfaces more general than the hyperplanes utilized in traditional projection pursuit. It is then expected that fewer numbers of projections are required (perhaps as few as $2d+1$).

13. Mathematical Foundations

Consider continuous functions $f(x_1, \ldots, x_d)$ of d variables on a bounded set such as the unit cube $[0,1]^d$. Upon reflection it appears that all familiar functions of three or more variables are built up from the composition of various functions of one or two variables. (For instance a sum of d variables is a composition of d-1 bivariate sums.) Accustomed to the traps of mathematical analysis, one might speculate that there exist truly d-dimensional functions that cannot be represented in this way. On the contrary, Kolmogorov (1957), see also Lorentz (1966), proved the surprising result that every continuous function on $[0,1]^d$ can be exactly represented as a composition of sums and continuous one-dimensional functions.

Lorentz (1966) identified a particular composition scheme (depicted in fig. 2) which works for all functions of a given dimension. For any continuous function f on $[0,1]^d$, there exist continuous one-dimensional functions g_j and h_{jk} for $j=1, 2, \ldots, 2d+1$ and $k=1, 2, \ldots, d$ such that

$$f(x_1, \ldots, x_d) = \sum_j g_j \Big(\sum_k h_{jk}(x_k) \Big) \qquad (8)$$

Moreover, Lorentz demonstrated the existence of universal functions h_{jk} which do not depend on the function f (whereas the g_j do depend on f). In his proof, Lorentz constructs piecewise linear functions $g_j^{(\varepsilon)}$ with the property that for every x in the cube the *majority* (i.e. at least $d+1$) of the values $g_j^{(\varepsilon)}(\sum h_{jk}(x))$ (for $j=1, \ldots, 2d+1$) are within ε of $f(x)$. (This proof suggests that it might be more natural to use the median of $g_1(\sum h_{1k}), \ldots, g_{2d+1}(\sum h_{2d+1,k})$ instead of the sum to approximate f.) The proof of the existence of an exact representation involves a careful limiting argument with $\varepsilon \to 0$.

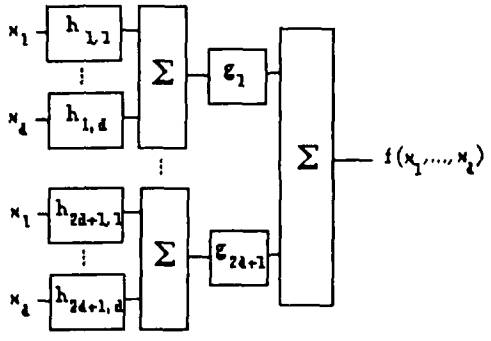


Fig. 7. Kolmogorov-Lorentz Network

In general the functions g_j for which the representation is valid may be rather irregular (e.g. nondifferentiable). It is reasonable to expect, that for sufficiently regular functions f, relatively smooth elements g_j and h_{jk} can be used in the representation, especially if the h_{jk} are allowed to depend on f.

One way to quantify the smoothness of a function is the characteristic s. A function of d variables has characteristic $s = p/d$, where $p = r + \alpha$ if all derivatives of order r are Lipshitz continuous of order α where $0 < \alpha \leq 1$ (this is the case with $\alpha = 1$, $r = p - 1$ if the derivatives of order p are bounded). (This smoothness characteristic is used by Stone (1982) to obtain minimax rates of convergence of nonparametric estimators, see below.) Kolmogorov (1959), see also Lorentz (1966), proved that *not every function with a given smoothness characteristic can be represented as a composition of functions*

having a larger smoothness characteristic. This means, for instance, that there exist functions of ten variables which are differentiable up to order ten that cannot be represented by compositions using one-dimensional functions having more than one derivative.

The limitations expressed by these theoretical results do not preclude the possibility that many of the practically occurring functions which one might wish to estimate are representable in terms of low-dimensional functions of large smoothness characteristic. For instance, it might be true that infinitely differentiable functions can be represented in terms of compositions of infinitely differentiable functions of low dimensionality.

The appeal of the Kolmogorov-Lorentz representation compared to other familiar network structures is the economy of network nodes. A *fixed* number of one-dimensional continuous functions (namely $(d+1)(2d+1)$) suffices to give an approximation or even an exact representation.

Other network structures are known to possess approximation capabilities, but generally the number of network nodes depends on the function being approximated and the desired accuracy. Subsequent to our *Interface* presentation, George Cybenko informed us of some of his recent results (Cybenko 1988). Consider three-layer networks in which the element in the final layer takes a linear combination of its inputs and the first two layers are restricted to elements in the form of equation (1), each of which uses the same nonlinear transformation h. This function h is permitted to be any fixed continuous strictly increasing function with bounded range. Cybenko proved that *for any continuous function f on a d-dimensional cube and any $\varepsilon > 0$, there exists a three-layer network with elements of the form (1) that approximates f with error uniformly less than ε.* His proof is to show that the first two layers of the network may be used to implement kernel functions ("approximations to the identity") of appropriate bandwidths having arbitrary centers, from which the result follows by taking an appropriate linear combination. Cybenko also points out that two-layer networks are sufficient if quadratic ϕ functions are used in first layer elements of the form (3), for then certain kernel functions may be constructed by taking linear combinations of these elements. Although Cybenko does not refer to the rich collection of statistical literature on kernel approximation (see the books by Prakasa Rao 1983, Devroye 1987, or Eubanks 1988), it is apparent that results in this area could be utilized to bound the number of kernels (and hence the number of nodes in Cybenko's networks) required to achieve a given accuracy.

Some basic results in mathematical analysis which have impact on the approximation capabilities of network forms should not be overlooked. The Weierstrass theorem and its generalization to multivariate functions asserts that any continuous function on $[0,1]^d$ can be uniformly approximated by a sufficiently large degree polynomial. The polynomial approximations need not be restricted to the canonical sum of products form $\sum \theta_k x_1^{k_1} \ldots x_d^{k_d}$ (which is itself a large network of simple structure), indeed, the multivariate generalization of Weierstrass's theorem is seen to be an immediate corollary to the Kolmogorov-Lorentz representation theorem.

Other multivariate forms are known to approximate arbitrary continuous functions. For instance, finite trigonometric sums $\sum_k (\alpha_k \cos(\pi k \cdot x) + \beta_k \sin(\pi k \cdot x))$ can uniformly approximate any continuous function on $[0,1]^d$, provided the function is continously extended to satisfy boundary conditions on $[-1,1]^d$ (see Lorentz 1966, p.87). Here $k = (k_1, \ldots, k_d)$ and $k \cdot x = \sum_j k_j x_j$. We remark that the sin and cos functions have bounded variation, so they can be represented as the difference of monotone functions h. Consequently, the trigonometric sum is a two-layer network with first layer elements having the form (1). This gives a simple proof of Cybenko's theorem specialized to such h.

The Jackson theorems express bounds on the accuracy of a polynomial or trigonometric approximation in terms of the assumed smoothness of the function being approximated. (See Jackson 1930 for a lucid treatment of the univariate case and Lorentz 1966, especially pp. 87-90, for multivariate extensions.) For instance, if a function f has partial derivatives $\partial^r f / \partial x_k^r$ of order $r \geq 0$ which are Lipshitz of order $0 < \alpha \leq 1$, then there is a constant c such that for every $N \geq 1$ a polynomial approximation of degree N (in each coordinate) exists with error uniformly less than cN^{-p}, where $p = r + \alpha$. Unfortunately, Jackson type theorems are not known for polynomial approximations which take a network form other than a sum of products.

14. Some Limitations on the Statistical Accuracy of Learning Networks

In practice, learning network approximations are not obtained from completely known functions, but rather they are estimated from a training sample of observations of relevant variables. The sample is typically a sequence of input/output pairs $X_1, Y_1, \ldots, X_n, Y_n$ which is assumed to possess one of several possible probabilistic structures as discussed previously. There is a fundamental question which is addressed for this class of problems: *What is the relationship between the achievable accuracy and the size n of the sample?* Typically it is found that the answer depends on the class of possible functions. Especially critical are the dimension d and the regularity of the function. Results from approximation theory play a key role in these statistical considerations. The presently known answers, which we discuss below, are somewhat discouraging, especially with regard to practical contraints imposed on the dimensionality. To understand better and to avoid the pitfalls of high dimensionality, it is suggested that new approximation theory and estimation results are needed for specific network composition strategies.

Stone (1982) has fundamental results concerning a class of nonparametric estimation problems which includes curve or surface fitting with normally distributed errors and binary classification with unknown conditional class probability functions. Attention is restricted to functions on a bounded set with a given smoothness characteristic $s = p/d$ (in the sense that all cross partial derivatives of total order r are Lipshitz of order α and $p = r + \alpha$ as above). Stone establishes that *the optimal rate of convergence is* $\varepsilon_n = n^{-s/(2s+1)}$ for the L^q norms $(0<q<\infty)$ *and* $\varepsilon_n = (n^{-1} \log n)^{s/(2s+1)}$ for the L^∞ norm. This means that there exist estimators $\hat{f}_n$ (depending only on the sample) such that the ratio $\| \hat{f}_n \| / \varepsilon_n$ is bounded in probability for all functions f of the given smoothness class. Conversely, for any sequence of estimators $\hat{f}_n$ there exist sequences of functions f of the given smoothness class for which the ratio $\| f - \hat{f}_n \| / \varepsilon_n$ is bounded away from zero in probability, as $n \to \infty$. To achieve the optimal rate of convergence, Stone (1982) uses *local polynomial regression*. The value of the estimator $\hat{f}_n(x)$ at a point x is obtained by a weighted least squares polynomial fit using all data points for which the distance from x is less than δ_n. Stone chooses the sequence δ_n to converge to zero at rate $n^{-1/(2p+d)}$ and he chooses the local polynomials to have total degree r.

For convergence of the mean integrated squared error (MISE) uniformly over all functions which have a bound on the L^2 norm of derivatives of order p, the optimal convergence rate is of the form $n^{-2p/(2p+d)}$. Indeed, a consequence of Stone's result is that this asymptotic rate cannot be improved. This rate is achieved in regression contexts by multivariate smoothing splines (Cox 1984) and in some cases by least squares polynomial regression and trigonometric series regression, see Cox (1988). A. R. Barron (1988) has analogous results for the

estimation of a log-density function. For the specia' case $d=1$, asymptotic (and in some cases exact) minimax estimators are found in Efroimovich and Pinsker (1983) for density estimation, and Nussbaum (1985) and Speckman (1985) for regression. In these univariate cases the constant $c(p)$ is determined in the asymptotic minimax error $c(p)n^{-2p/(2p+1)}$. For $d>1$, it appears that the corresponding constant c(p,d) for exact asymptotics $c(p,d)n^{-2p/(2p+d)}$ is not yet explicitly determined. Determination of the behavior of this constant for large d would be useful, since it would help determine whether practical minimax estimation is possible in high dimensions.

Observe that unless the degree of smoothness p is large compared to the dimension d, the optimum rate of convergence $n^{-p/(2p+d)}$ is disappointingly slow. For instance, with dimension $d = 8$ and smoothness $p = 2$, a sample of size $n \geq 10^6$ (one million!) would be required to make $n^{-p/(2p+d)}$ be not greater than 1/10.

The slow rates for optimal estimation of smooth functions in high dimensions suggest that to understand the practical success of certain high-dimensional estimation strategies it may be necessary to use notions of the regularity of a function other than differentiability to quantify the limits on statistical accuracy. One possibility is to assume proximity of the desired function to functions of low Kolmogorov complexity. It may then be possible to obtain rate of convergence results as well as the consistency results referred to in section 5 (for networks selected by complexity regularization). This is a topic of further investigation.

In recent work by Baum and Haussler, the Vapnik-Chervonkis dimension of families of network functions is characterized and used to quantify the statistical reliability of estimated networks for binary classification. Using results of Cover (1965, 1967) on the number of possible dichotomies of a sample by networks of thresholded linear elements, Baum (1988) has bounded the Vapnik-Chervonkis dimension in terms of the total number of coefficients in the network. Let $0<\varepsilon_1<\varepsilon_2<1$ be given. Suppose it is observed that the fraction of errors of an estimated network is less than ε_1 on a training sample of size n. Then it is of interest to bound the conditional probability that a fraction of at least ε_2 errors will be incurred by this network on an independent test sample. Baum and Haussler (1988) have some results in this direction, assuming that the total number of coefficients is sufficiently small compared to the sample size.

The advantage of the Baum and Haussler approach is its uesfulness in retrospective analysis: i.e., *given* that an accurate estimate has been found on training data, what is the probability of error likely to be on new data? This approach avoids questions concerning the approximation capabilities of a network: in particular, the probability that an estimated network will achieve a certain accuracy is not determined.

15. Conclusions

Historically, neural networks, adaptive polynomial learning, and nonparametric statistical inference are fields of inquiry with distinct perspectives and separate lines of development which have crossed paths only on occasion. However, by examining the purpose, scope, and methodologies in these fields, considerable commonality is revealed. In each case, network functions are used to approximate possibly complex multivariate relationships by composition of many simpler relationships. Moreover, strategies for the synthesis of these networks from observable data are developed. To understand the performance of these strategies and to suggest improved methodologies, practical experience is supplemented by an understanding of the basic disciplines of mathematical approximation theory and statistical decision theory. Conversely, it behooves the practitioner in multivariate nonparametric statistical inference to become aware of the benefits and experiences in the use of multiple-layered networks for classification, regression, and related problems.

In our experience the most successful learning network methodologies adaptively grow the network structure, using all the observational data (in batch rather than recursively) and using an appropriate model selection criterion to ensure a parsimonious network. Moreover, the best strategies employ network structures which are not limited in their approximational capabilities. The principle examples of these successful methodologies are adaptively synthesized polynomial networks and projection pursuit.

It appears to us that several different approaches lead inevitably to one network structure and similar synthesis strategies: namely the network of fig. 7 (introduced by Kolmogorov and Lorentz) estimated by a generalization of projection pursuit which incorporates additive projections or estimated by polynomial network strategies specialized to this structure. This network considerably extends the capabilities of existing projection pursuit and additive regression models, yet retains enough of the regularity of these models that it may be amenable to further theoretical and practical examinations of its properties. Nevertheless, we should not restrict all attention to just one network structure. Hopefully, by consideration of a variety of different compositions, empirically selecting the best (say by complexity regularization), discovery of the true relationships can occur.

Appendix: Convergence of networks estimated by complexity regularization

In this appendix we specialize some results from A.R. Barron (1985,1987) to show convergence of estimates of network functions. In general the theory is concerned with the selection of a probability distribution using random data $W^n = (W_1, W_2, \ldots, W_n)$. It is assumed that Γ is a countable collection of probability distributions which are candidates for the estimate of the distribution of the process $W_1, W_2, \ldots$ and that $L(P)$, $P \in \Gamma$ are positive numbers which satisfy the Kraft-McMillan inequality $\sum_{P\in\Gamma} 2^{-L(P)} \leq 1$. (Here $L(P)$ may be regarded as the length of a uniquely decodable code or $2^{-L(P)}$ may be regarded as a discrete prior probability.) Short lengths $L(P)$ are desired for as large as possible a set of distributions that can be computed, so ideally, we would let $L(P)$ be the Kolmogorov complexity (relative to a fixed universal computer) and Γ would be the set of all computable distributions; however, the determination of such an ideal complexity is practically infeasible. Nevertheless, the complexity principle provides a useful guide in selecting reasonable sets of distributions and assigning priors geared toward parsimonious distributions. When the distribution is known except for a function f of d variables on which the distribution P_f depends, then families of network functions and corresponding description lengths can be used to yield an effective criterion for selecting an appropriate network.

In general the *complexity regularization* estimator $\hat{P}_n$ is defined to achieve

$$\min_{P\in\Gamma}\{-\log p^n(W_1,\ldots,W_n) + L(P)\} \qquad (9)$$

Here the density functions p^n are taken with respect to a fixed dominating measure. Logarithms are taken base 2. When $W_1,\ldots,W_n$ are discretized random variables, then $-\log p(W_1,\ldots,W_n)$ (upon rounding up to the nearest integer) is the length of a Shannon code for these variables based on the distribution P and the term $L(P)$ is the length of a preamble required to specify which distribution. A more general form of complexity regularization is to minimize

$$CR = -\log p^n(W^n) + \lambda L(P) \qquad (10)$$

where λ may be regarded as a Lagrange multiplier. Unless $\lambda = 1$, CR does not have the same total description length interpretation. Nevertheless, the solutions $\hat{P}_n$ which minimize CR for $\lambda > 0$ do have the valid interpretation as maximum likelihood estimators subject to complexity constraints. Such estimators were first proposed by Cover (1972). Our convergence results require that $\lambda \geq 1$ be fixed, although in one case $\lambda > 1$ is required.

We mention several general convergence results. First suppose that the distributions P in Γ are stationary and ergodic. Let P^* denote the true probability law which governs the process. The first result is that *if $P^* \in \Gamma$ then the estimated distribution is exactly correct, $\hat{P}_n \equiv P^*$, for all large n, with probability one.* For the remaining results suppose that the variables W_i are independent and identically distributed with respect to P^*, and likewise that independence holds for the distributions in Γ, whence $p(W_1,\ldots,W_n) = \prod p(W_i)$. Moreover, it is assumed that the true density function p^* can be approximated by densities in Γ in an information theoretic sense: that is, there exist densities in Γ for which the relative entropy $\int p^* \log p^*/p$ is arbitrarily small. This leads to the second result that $\hat{P}_n \Rightarrow P^*$ (in the sense of weak convergence) with probability one; moreover, if the densities in Γ are uniformly equicontinuous then $\hat{p}_n \rightarrow p^*$ in L^1. Since the uniform equicontinuity is not easy to guarantee in general, we mention a third result which makes no such requirement. *If $\lambda > 1$ and if densities in Γ can approximate p^* in the relative entropy sense, then $\hat{p}_n \rightarrow p^*$ in L^1, that is* $\lim \int |\hat{p}_n - p^*| = 0$, *with probability one.* The second and third results continue to be valid (with convergence in probability statements replacing convergence with probability one) when the set Γ_n and the numbers $L_n(P)$ are allowed to depend on the sample size n, provided that there exists a sequences of densities p_n in Γ_n for which $\lim \int p^* \log p^*/p_n = 0$ and $\lim L_n(P_n)/n = 0$.

For the estimation of network functions we take $W_i = (\underline{X}_i, Y_i)$ which is assumed to have a distribution P_f which depend on the function we desire to estimate. A denumerable (possibly finite) collection S_n of parameterized families of network functions $f(x,\theta)$ is considered. We assume that the sequence of collections is increasing $S_1 \subset S_2 \subset \cdots$ and that $L(f)$, $f \in S$ are lengths of codes which specify the structure, but not the parameter values, of networks in $S = \bigcup_n S_n$. For each network family f, the parameter vector (which has dimension denoted by k_f), is assumed for convenience to take values in the unit cube $[0,1]^{k_f}$. (Families with larger rectangular parameter spaces can be reduced to this case by scaling and appropriately modifying the definition of f). We restrict attention to the lattice $\Omega_{n,k}$ of points with coordinates of the form i/n for integers $0 \leq i < \sqrt{n}$ and we use $(1/2)\log n$ bits per parameter to describe these points.

For each parametrized network $f(x,\theta)$ in S_n, let $\hat{\theta}_n$ be estimated by the method of maximum likelihood restricted to the parameter values of the given precision. Thus $\hat{\theta}_n$ achieves

$$p(W^n | f(\cdot,\hat{\theta}_n)) = \max_{\theta \in \Omega_{n,k_f}} P(W^n | f(\cdot,\theta)). \qquad (11)$$

The complexity regularization estimator is the network $\hat{f}_n$ defined to achieve

$$\min_{f \in S_n} \{-\log p(W^n | f(\cdot,\hat{\theta}_n)) + \lambda \frac{k_f}{2} \log n + \lambda L(f)\}. \qquad (12)$$

We remark that other precisions than $(1/2)\log n$ bits could be used in the definition, provided the maximum likelihood estimator is suitably restricted. (For smooth families, a second order Taylor series argument shows that the present choice achieves roughly the best tradeoff between complexity and likelihood. In some cases an improved tradeoff is obtained using local reparametrizations as dictated by the Fisher information matrix, as in A.R. Barron (1985, p. 74). With $\lambda = 1$, the specialization of the complexity regularization criterion given in (12) is very much the same as Rissanen's MDL criterion. However, the $L(f)$ term (omitted by Rissanen) can be important, especially when there is a large variety of families under consideration.

As a special case of interest consider function fitting problems with Gausssian errors. In this case, for given $\underline{X}$, the conditional distribution of the error $Y - f(\underline{X})$ is normal with mean zero and variance σ^2. The $\underline{X}_i$ are assumed to be randomly selected, independently, from a distribution which does not depend on f. Then the complexity regularization criterion reduces to

$$CR = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - f(X_i,\hat{\theta}))^2 + \lambda \frac{k_f}{2} \log n + \lambda L(f). \qquad (13)$$

Let f^* be the true function which we desire to estimate. Assuming the the network in S are continuous functions of their parameters, the information theoretic closure condition reduces (in the Gaussian case) to the condition that $\inf_{f \in S} \inf_\theta E(f^*(X) - f(X,\theta))^2$, i.e. the true function must be approximable in the L^2 sense by members of network families under consideration. In which case, networks $\hat{f}_n(X)$ which are selected to minimize (13) (with $\lambda > 1$) are guaranteed to converge to $f^*(X)$ in probability.

References

H. Akaike (1970) Statistical predictor identification, *Ann. Inst. Stat. Math*, 22 203-217.

H. Akaike (1973) Information theory and an extension of the maximum likelihood principle, *Proc. 2nd Int. Symp. Inform. Theory*, B.N. Petrov and F. Csaki (Ed.s), 267-281 Akademia Kiado, Budapest.

J.A. Anderson and E. Rosenfeld (1988) *Neurocomputing: Foundations of Research* MIT Press.

S. Agmon (1954) The relaxation method for linear inequalities, *Canadian J. Math*, 6 382-392,

U.K. Ashby (1952) *Design for a Brain*, Wiley, New York.

A.C. Atkinson (1980) A note on the generalized information criterion for choice of a model, *Biometrica*, 67 413-418.

A.C. Atkinson (1981) Likelihood ratios, posterior odds and information criteria, J. *Econometrics*, 16 15-20.

A.R. Barron, F.W. van Straten, and R.L. Barron (1977) Adaptive learning network approach to weather forecasting: a summary, *Proc. IEEE Int. Conf. Cybernetics and Society*, 724-727.

A.R. Barron (1984) Predicted squared error: a criterion for automatic model selection, *Self-Organizing Methods in Modeling*, S.J. Farlow (Ed.), Marcel Dekker, New York.

A.R. Barron (1985) *Logically Smooth Density Estimation*, Ph.D. Thesis, Stanford University.

A.R. Barron (1987) The exponential convergence of posterior probabilities with implications for the consistency of Bayes density estimators, submitted to *Ann. Statist.*

A.R. Barron (1988) Approximation of densities by sequences of exponential families, *submitted to Ann. Statist.*

R.L. Barron, R.F. Snyder, E.A. Torbett, and R.J. Brown, (1964) *Advanced Computer Concepts for Intercept Prediction*, Adaptronics, Inc. Final Technical Report, Army Nike-X Project Ofc., Redstone Arsenal, AL, November 1964. (See Vol.I: *Conditioning of Parallel Networks for High-Speed Prediction of Re-entry Trajectories.*)

R.L. Barron (1974) Theory and application of cybernetic systems: an overview, *Proc. 1974 IEEE Nat. Aerospace and Elect. Conf.*, 107-118.

R.L. Barron (1975) Learning networks improve computer-aided prediction and control, *Computer Design*, August 1975, 65-70.

R.L. Barron, A.N. Mucciardi, F.J. Cook, J.N. Craig, and A.R. Barron (1984) Adaptive learning networks: development and application in the United States of algorithms related to GMDH, *Self-Organizing Methods in Modeling*, S.J. Farlow (Ed.), Marcel Dekker, New York.

R.L. Barron and D. Abbott (1988) Use of polynomial networks in optimum, real-time, two-point boundary-value guidance of tactical weapons, *Proc. Military Comp. Conf.*, May 3-5, Anaheim, CA.

E.B. Baum (1988) On the capabilities of multilayer perceptrons, *J. Complexity* (in press).

E.B. Baum and D. Haussler (1988) What size net gives valid generalization?, *IEEE Int. Symp. Inform. Theory*, June 19-24, Kobe, Japan.

L. Breiman and J.H. Friedman (1985) Estimating optimal transformations for multiple regression and correlation (with discussion), *J. Am. Stat. Assoc.*, 80 580-619.

T.M. Cover (1965) Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Trans. Elect. Comp.*, 326-334.

B. Clarke and A.R. Barron (1988) Information theoretic asymptotics of Bayes methods, submitted to *IEEE Trans. Inform.*

T.M. Cover (1967) Capacity problems for linear machines, *Statistical Classification Procedures*, 283-289.

D.D. Cox (1984) Multivariate smoothing spline functions, *SIAM J. Numer. Anal.*, 21 789-813.

D.D. Cox (1988) Approximation of linear regression on nested subspaces, *Ann. Stat.*,18.

G. Cybenko (1988) *Continuous Valued Neural Networks with Two Hidden Layers are Sufficient*, Tech. Report Dept. Computer Science, Tufts Univ. Medford, Mass.

L. Devroye (1987) *A Course in Density Estimation*, Birkhauser, Boston, Mass.

D. Donoho and I.M. Johnstone (1985) Discussion on projection pursuit, *Ann. Stat.*, 13 496-500.

R.O. Duda and P.E. Hart (1973) *Pattern Classification and Scene Analysis*, Wiley, New York.

R.L. Eubanks (1988) *Spline Smoothing and Nonparametric Regression*, Marcel Dekker, New York.

S.Y. Efroimovich and M.S Pinsker (1982) Estimation of square-integrable probability density of a random variable, *Problems in Information Transmission*, 18 175-189.

B. Efron (1964) The perceptron correction procedure in nonseparable situations, *Rome Air Development Center Technical Documentary Report*, RADC-TDR-63-533.

B. Farley and W. Clark (1954) Simulation of self-organizing systems by digital computer, *IRE Trans. Inform. Theory*, 4 76-84.

R.A. Fisher (1922) The goodness of fit of regression formulae and the distribution of regression coefficients, *J. Roy. Stat. Soc.*, 85 597f.

R.A. Fisher (1934) Probability likelihood and quantity of information in the logic of uncertain inference, *Proc. Roy. Soc. A*, 146 1f.

R.A. Fisher (1936) The use of multiple measurements in taxonomic problems, *Ann. Eugenics*, 7 179-188

J.H. Friedman and J.W. Tukey (1974) A projection pursuit algorithm for exploratory data analysis, *IEEE Trans. Computers*, 23 881-889.

J.H. Friedman and W. Stuetzle (1981) Projection pursuit regression, *J. Amer. Stat. Assoc.*, 76 817-823.

J.H. Friedman, W. Stuetzle and A. Schroeder (1984) Projection pursuit density estimation, *J. Amer. Stat. Assoc.*, 79 599-608.

L.O. Gilstrap Jr. (1971) Keys to developing machines with high-level artificial intelligence, *Proc. ASME Design Eng. Conf.*, ASME Paper 71-DE-21.

D.O. Hebb (1949) *The Organization of Behavior*, Wiley, New York.

J. Hopfield (1982), Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Ac. of Sciences*, 79 2554-2558.

P.J. Huber (1985) Projection pursuit (with discussion), *Ann. Stat.*, 13 435-525.

A.G. Ivakhnenko (1971) Polynomial theory of complex systems, *IEEE Trans. Systems, Man, Cybernetics*, 1 364-378.

D. Jackson (1930) *The Theory of Approximation*, Am. Math. Soc., New York.

L. Jones (1986) Convergence of generalized projection pursuit in the nonsampling case, unpublished manuscript.

L. Jones (1987) On a conjecture of Huber concerning the convergence of projection pursuit regression, *Ann. Stat.*, 15 880-882.

A.N. Kolmogorov (1957) On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition, *Dokl*, 114 679-681.

A.N. Kolmogorov and V.M. Tikhomirov (1959) Entropy and ε-capacity of sets in function spaces, *Uspehi*, 14 3-86.

R.J. Lee and L.O. Gilstrap (1960) Learning machines, *Proc. Bionics Symp., USAF Wright Air Development Division, Dayton, Ohio, TR60-600* 437-450.

G.G. Lorentz (1976) The 13th problem of Hilbert, in *Mathematical Developments Arising from Hilbert Problems*, F.E. Browder (Ed.), Am. Math. Soc., Providence, R.I.

C.L. Mallows (1973) Some comments on Cp, *Technometrics*, 15 661-675.

M.L. Minsky (1954) *Neural-Analog Networks and the Brain Model Problem*, Ph.D. Thesis, Princeton Univ.

M.L. Minsky and S. Papert (1969) *Perceptrons: An Introduction to Computational Geometry*, M.I.T. Press, Cambridge, Mass.

W.S. McCulloch and W. Pitts (1943) A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophysics*, 5 115-133.

R.E.J. Moddes , R.J. Brown, L.O. Gilstrap Jr., et al. (1965) Study of Neurotron Networks in Learning Automata, Adaptronics Inc., Air Force Avionics Laboratory, Dayton, Ohio, AFAL-TR-65-9.

A.N. Mucciardi (1972) Neuromime nets as the basis for the predictive component of robot brains, *Cybernetics, Artificial Intelligence, and Ecology*, H.W. Robinson and D.E. Knight (Eds.), Spartan Books, 159-193, 4th Ann. Symp. Am. Soc. Cybernetics, October 1970.

N.J. Nilsson (1965) *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, New York.

J. von Neumann (1956) Probabilistic logics and the synthesis of reliable organisms from unreliable components, *Automata studies*, C.E. Shannon and J. McCarthy (eds), Princeton Univ. Press, 43-98.

M. Nussbaum (1985) Spline smoothing in regression models and asymptotic efficiency in L2, *Ann. Stat.*, 13 984-997.

Prakasa Rao (1983) *Nonparametric Functional Estimation*, Academic Press, Orlando.

J. Rissanen (1978) Modeling by shortest data description, *Automatica*, 14.465-471.

J. Rissanen (1983) A universal prior for integers and estimation by minimum description length, *Ann. Stat.*, 11 416-431.

J. Rissanen (1984) Universal Coding, Information, Prediction, and Estimation, *IEEE Trans. Inform. Theory*, 30 629-636.

H.E. Robbins and Monroe (1951) A stochastic approximation method, *Ann. Math. Stat.*, 22.400-407.

F. Rosenblatt (1958) *The Perceptron. A Theory of Statistical Separability in Cognitive Systems*, Cornell Aeronautical Laboratory Report No. VG-1196-G-1.

F. Rosenblatt (1962) *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, D.C.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams (1986) Learning representations by back propagations, *Nature* , 323 533-536.

D.E. Rumelhart and J.L. McClelland (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol.1: *Foundations*, M.I.T. Press, Cambridge, Mass.

G. Schwarz (1978) Estimating the dimension of a model, *Ann. Stat.*, 6 461-464.

R. Shibata (1984) Approximate efficiency of a selection procedure for the number of regression variables, *Biometrika*, 71 43-49.

R. Shibata (1986) Selection of the number of regression variables: a minimax choice of generalized FPE, *Ann. Inst. Stat. Math.*, 38 459-474.

S. Shrier, R.L. Barron, and L.O. Gilstrap (1987) *Proc. IEEE 1st Int. Conf. Neural Networks* II, 431-439.

P. Speckman (1985) Spline smoothing and optimal rates of convergence in nonparametric regression models, *Ann. Stat.*, 13 970-983.

C.J. Stone (1982) Optimal global rates of convergence for nonparametric regression, *Ann. Stat.*, 10 1040-1053.

C.J. Stone (1985) Additive regression and other nonparametric models, *Ann. Stat.*, 13 689-705.

R. Tibshirani (1988) Estimating transformations for regression via additivity and variance stabilization, *J. Am. Stat. Assoc.*, 83 394-405.

B. Widrow and M.E. Hoff (1960) Adaptive switching circuits, *1960 IRE WESCON Convention Record*, 96-104.

B. Widrow (1962) Generalization and information storage in networks of Adaline neurons, *Self-Organizing Systems*, M.C. Yovits, G.T. Jacobi, and G.D. Goldstein (ed's), Spartan Books, Washington, D.C., 435-461.

B. Widrow, R.G. Winter, and R.A. Baxter (1987) Learning phenomenona in layered neural networks, *Proc. IEEE 1st Int. Conf. Neural Networks II*, 441-430.

S. Winsberg and J.O. Ramsay (1980) Monotonic transformations to additivity using splines, *Biometrica*, 67 669-674.

MARKOV CHAINS ARISING IN COLLECTIVE COMPUTATION NETWORKS WITH ADDITIVE NOISE

Robert H. Baran, Naval Surface Warfare Center

ABSTRACT

Recent progress in modelling connectionist ("neural") networks gives rise to the expectation that future computing systems will employ coprocessors in which large numbers of memoryless, nonlinear processing units interact through plastic connections. Hopfield has drawn attention to symmetrically interconnected networks of binary threshold units. These collective computation networks converge rapidly to stable states corresponding to local minima of the computational energy. The network can be freed from local minima by the addition of noise at the input of each neuron-like unit. The state then takes a random walk on the 2^N vertices of a hypercube, where N is the number of "neurons". This paper uses a simple, explicit algorithm to study the behavior of collective computation networks with additive noise. The algorithm gives rise to a stationary Boltzmann distribution of the network state. Formulas for the temperatures of non-logistic noises are derived and tested in Monte Carlo trials.

INTRODUCTION

This concerns one of the folk theorems of statistical neurodynamics, which holds that the states of a globally asymptotically stable neural network, subjected to isothermal agitation, occur with relative frequencies given by the Boltzmann distribution. Global asymptotic stability follows from the existence of an energy function H of the networks state $\mathbf{s}$. This was discovered by Hopfield [3], whose expression

$$H(\mathbf{s}) = -\sum_{j=i+1}^{N}\sum_{i=1}^{N-1} S_i S_j T_{ij} - \sum_{i=1}^{N} S_i U_i \qquad (1)$$

for the computational energy of the network is analogous to the Hamiltonian of a collection of interacting magnetic dipoles. Here S_i is the state of the i-th neuron-like element--either firing at its peak rate (S_i=1) or resting (S_i=0); and the connectivity matrix $||T_{ij}||$ gives the strength of the "synapse" through which the i-th unit excites (if $T_{ij}>0$) or inhibits (if $T_{ij}<0$) the j-th unit. The state of the i-th unit is decided by a threshold test applied to its input,

$$X_i = \sum_{j=1}^{N} S_j T_{ji} + U_i. \qquad (2.1)$$

The binary "McCulloch-Pitts neuron" obeys the rule

$$S_i = \begin{cases} 1 & \text{if } X_i > 0 \\ 0 & \text{if } X_i \le 0. \end{cases} \qquad (2.2)$$

When the T-matrix is real-valued and symmetric, with all zeros on the diagonal, the network evolves toward stable states which correspond to local minima of the computational energy. The "energy landscape" can be configured so that these local minima correspond to solutions of constrained optimization and pattern recognition problems [7,8]. In the latter case, the vector $\mathbf{U}$ of inputs to the N units might represent the pixel pattern on a retina.

"BOLTZMANN MACHINES"

A provocative paper by Ackley, Hinton and Sejnowski [1] proposed simulated annealing to dislodge the Hopfield network from local minima and enable it to settle into states of still lower energy which would represent better (if still suboptimal) solutions. The network is "heated" by the addition of noise to the input of each unit. When these noises are independent, identically distributed random variables, the state $\mathbf{s}$ takes a random walk on the 2^N vertices of a hypercube. The stationary distribution is

$$\Pr(\mathbf{s} = \mathbf{s}) = \exp[-\beta H(\mathbf{s})] / \sum_{\mathbf{s}'} \exp[-\beta H(\mathbf{s}')]. \qquad (3)$$

The assertion of Ackley, Hinton and Sejnowski, that $1/\beta = T$ is the root mean intensity of noise described by a logistic distribution, was not powerfully motivated. Shaw et. al. [6] had earlier arrived at an expression like (3) in which β is a "smearing factor" determined from details of a stochastic model of the chemical synapse.

It was over a hundred years ago that Gibbs sought time-invariant solutions to a Liouville equation in which the independent variables were the Hamiltonian coordinates of a multiparticle system and the dependent variable was the probability of the system being in a given state. He arrived at a canonical ensemble in which "the index of probability [ie., the log-probability] is a linear function of the energy" of the state. This result is expressed by equation (3), called a Boltzmann distribution. Other functions of the energy, however, will serve this

purpose; and the fact that the linear dependence (of log-probability on energy) maximizes the entropy of the system is not necessarily germane to the question. Belief in the possibility of a mathematical treatment of biological intelligence, patterned after statistical thermodynamics, goes back at least as far as the works of John Von Neumann, published posthumously. For this belief to find expression in contemporary neural network research is not surprising. The mathematician who studies this work must be slightly bewildered by derivations which appeal to analogies with statistical physics, some of which are complicated by psychological theory [5]. The validity of the Boltzmann distribution, in the context of the connectionist paradigm, is solely dependent on the existence of models which give rise to it. As far as real neuronal networks are concerned, the laboratory experiments which would verify the result have yet to be defined.

The Algorithm

The computational technique of simulated annealing traces its roots to the Metropolis [4] algorithm, which updates the state of an N-particle system according to a stochastic model in which the Boltzmann distribution is expressly assumed beforehand. An alternative derivation due to C. R. Darnafalski, the amateur mathematician whose unpublished essays have been cited elsewhere [2], involves the following stochastic model: Pick an integer $i \in \{1,\ldots,N\}$ at random. Compute X_i according to (2.1). Modify X_i by the addition of a real random variable, call it Y_1, which is symmetrically distributed about a mean of zero. Compute S_i according to (2.2). These steps are iterated indefinitely with independent, identically distributed random numbers $\{Y_k,\ k=1,2,\ldots\}$. It is not hard to see that this gives rise to a sequence $\{\boldsymbol{s}_k,\ k=1,2,\ldots\}$ of states which constitute a Markov chain. Nonzero probabilities are attributed to transitions which involve at most one component of the state vector. With no external input ($\boldsymbol{U} = \boldsymbol{0}$), these probabilities depend on the T_{ij} and the distribution of Y, as described in the Appendix. When Hopfield's conditions are obeyed by the former, the stationary distribution can be derived analytically. This distribution is

$$\Pr(\boldsymbol{S} = \boldsymbol{s}) = Z^{-1} \exp\left(\sum_{j=i+1}^{N} \sum_{i=1}^{N-1} s_i s_j \log(F(T_{ij})/[1-F(T_{ij})]) \right), \quad (4)$$

in which F is the (cumulative) distribution function of Y and the denominator Z is the sum over all states which normalizes the discrete density. The assumption of logistic noise, as

$$F(y) = 1/(1 + e^{-\beta y}), \quad -\infty < y < \infty, \quad (5)$$

gives the last equation a particularly simple form (3).

Asymptotic Temperature

With regard to (4), suppose that the root mean intensity of the noise is large compared to each $T_{ij} = t$. Then the first order Taylor series expansion of the logarithm is

$$\log(F(t)/[1-F(t)]) = 4tF'(0)$$

since $F(0)=1/2$. Defining the asymptotic temperature T_∞ of the network in such a manner that $\beta = 1/T_\infty$ in (3), we shall have

$$T_\infty = 1/[4f(0)] \quad (6)$$

in terms of the probability density $f(y) = F'(y)$. When (5) is assumed, the last equation is indeed valid for $\beta = 1/T$. If the noise were normally distributed with standard deviation σ, then the asymptotic temperature would be

$$T_{NORMAL} = (\pi\sigma/2)^{1/2}/2.$$

If the noise had a Cauchy density $f(y) = (1/\pi)c/(c^2+y^2)$, the temperature would be

$$T_{CAUCHY} = \pi c/4.$$

Clearly this asymptotic temperature is not a function of the mean noise intensity, since the variance of the Cauchy random variable is undefined.

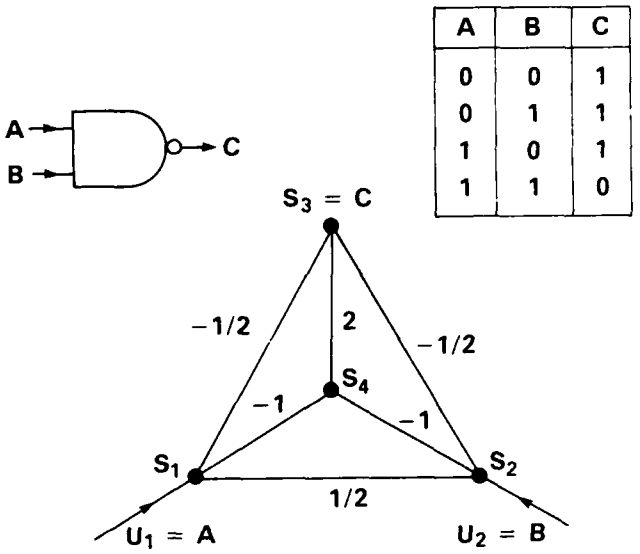
SIMULATIONS

Figure 1 represents a Hopfield net of four units in which the labeled segments give the dimensionless strengths of the symmetric interconnections. Let the inputs to units 1 and 2 be denoted A and B, respectively; and let $S_3 = C$. We shall consider only binary {0,1} inputs. The insets suggest that this small network performs the NAND (Not-AND) logic function C(AB) which the truth table (right) defines. This would indeed be the case if the network always settled into the state which gives the global or absolute minimum energy. Table 1 uses the formula

$$m(\boldsymbol{s}) = \sum_{i=1}^{4} s_i 2^{i-1}$$

to assign a natural number m to each of the 16 states of the network; and it lists (-1 times) the energies of the states for each input condition AB $\in$ {00, 01,10,11}. With input AB=11, the minimum energy is -2.5 and it occurs in state m=3 for which C is zero. With the other inputs, the minimum energy is -2.0 and occurs in state m=12 for which C=1. This motivates the truth table of Figure 1.

Figure 2 is a state transition map to



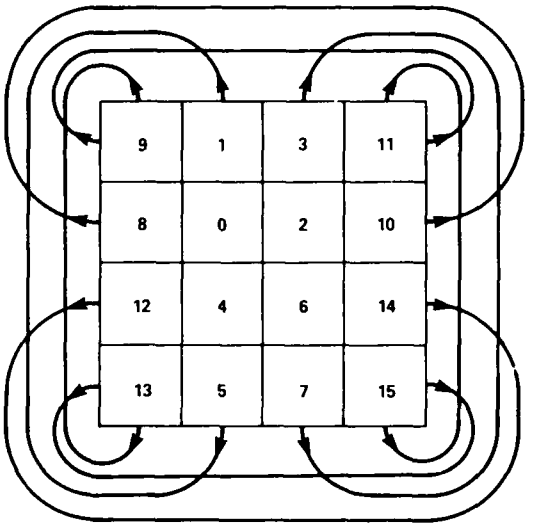
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Figure 1. A Hopfield net of four units, two of which receive binary inputs (A and B) and one of which registers the output (C). The T-matrix is specified by the labels on the line segments linking the units. This net is designed so that, for given inputs, the global minimum energy state gives a functional dependence C(A,B) as shown in the truth table (inset upper right), which defines the Not-AND (NAND) logic function.

show which transitions are allowed. Since units are interrogated in a random serial order, only one unit can toggle at a time. Thus the allowed transitions are of Hamming distance one. The 16 states of the "NAND gate" correspond to squares in the 4x4 array of the map. The squares are labeled with the values m(**s**). Motion is horizontal or vertical--never diagonal --between adjacent squares. The map wraps around horizontally and vertically as indicated by the connecting lines and arrows.

The interaction map of Figure 3 consists of four sub-maps each with the structure of the preceding Figure. Here each square is labeled with $-H_m(AB) = -H(\mathbf{s}[m],\mathbf{U}[AB])$. The four sub-maps correspond to the four input conditions. If the network begins in state m=3 with AB=11, the energy is minimized and the state is stable. Now if the input changes, the network is unable to leave the initial state, because any allowed transition will increase the energy. Similarly, if the initial state is m=12, and the input is subsequently set to AB=11, the state cannot assume the desired value (m=3) except by way of intermediate states of higher energy.

When noise is injected into the units of the network, the state can be dislodged from local (and global) energy minima. The Boltzmann distribution of



$s_1 s_2$ C s_4	DECIMAL STATE
0 0 0 0	0
1 0 0 0	1
0 1 0 0	2
0 0 1 0	4
0 0 0 1	8
ETC.	⋮

Figure 2. The state transition map for the network of Figure 1 uses the indicated binary-to-decimal convention to assign an integer (0 through 15) to each state of the net. Each square represents a state. Allowed transitions, which are of Hamming distance one, correspond to vertical or horizontal motion from one square to one of four adjacent squares.

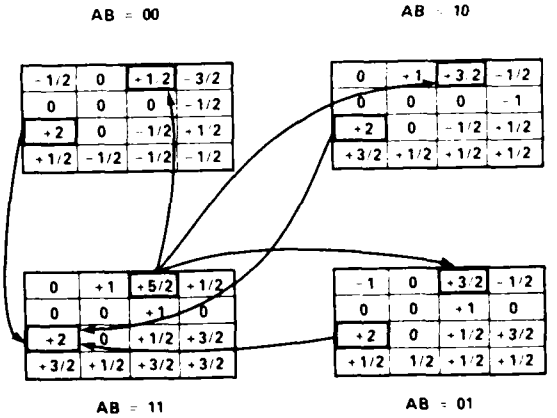


Figure 3. The interaction (negative energy) maps for each of the four input conditions have the same format as Figure 2; but the squares are labled with -1 times the computational energies. Arrows emphasize the entrapment of the four unit "NAND gate" in local energy minima.

Table 1. Interaction values of the sixteen states of the four unit NAND gate indicating global maxima for each input condition.

	AB=			
STATE	00	10	01	11
0	0	0	0	0
1	0	1	0	1
2	0	0	1	1
(3)	-.5	1.5	1.5	(2.5)
4	0	0	0	0
5	-.5	.5	-.5	.5
6	-.5	-.5	.5	.5
7	-.5	.5	.5	1.5
8	0	0	0	0
9	-.5	0	-1	0
10	-.5	-1	0	0
11	-1.5	-.5	-.5	.5
12	2	2	2	2
13	.5	1.5	.5	1.5
14	.5	.5	1.5	1.5
15	-.5	.5	.5	1.5

the network state is indeed observed in Monte Carlo trials with the network of Figure 1, to an accuracy consistent with sample size. Figure 4 shows the results of one such test in which 999 observations of **s** were recorded at random intervals in the course of ten thousand iterations of the algorithm described above. Here the input is AB=00 so that the modal probability (ie., the probability of the most likely state) is $p_{12} = Pr(m[\mathbf{s}]=12)$. This test used logistic noise with temperature T=1.

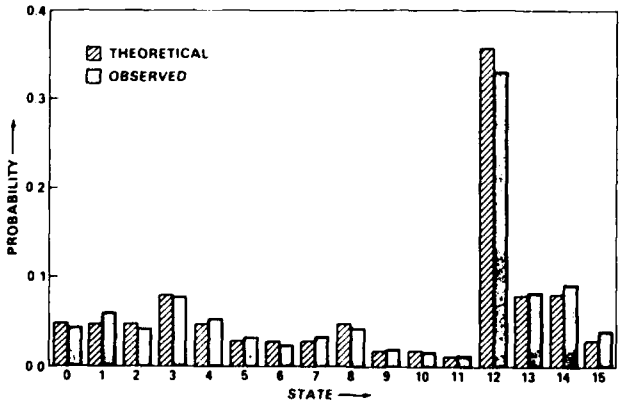


Figure 4. Theoretical and observed distributions of the network state with logistic noise at a temperature T=1. Sample distribution is based on 999 observations.

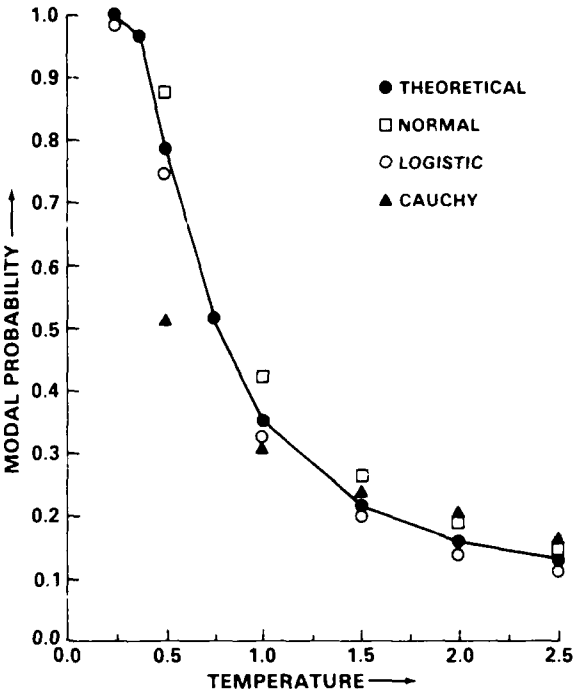


Figure 5. Modal probability versus temperature for the four unit "NAND gate" with zero input using three kinds of noise.

When the noise is not logistic, deviations from the Boltzmann distribution are apparent, especially at lower temperatures. Figure 5 shows the variation of the modal probability with temperature for each of three noise distributions.

ANALYSIS AND CONCLUSION

One measure of the disparity of two discrete probability densities, **p** and **q**, is the directed divergence, or (Kullback) information for discrimination against **p** in favor of **q**:

$$I(\mathbf{q},\mathbf{p}) = \sum_m q_m \log(q_m/p_m).$$

It is well known that, if **q** is a sample distribution, obtained from J independent observations of a random variable with discrete density **p**, $p_m > 0$ for all $m \in (0,\ldots M-1)$, then the product $JI(\mathbf{q},\mathbf{p})$ is chi-square with M-1 degrees of freedom in the limit $M/J \longrightarrow 0$. Then the mean value of the product JI is approximately M-1 for large J; and values of JI in obvious excess of M-1 will tend to refute the null hypothesis **p**.

Table 2 shows the product JI of the sample size and the discrimination information with the Boltzmann distribution as the null hypothesis. Each point represents about a thousand observations of the state of the four unit "NAND gate" at

random intervals in the course of runs of length 10,000. Three different noise distributions are considered with the input AB=00 at each of five temperatures. The expected value of the statistic is M-1 = 15 if the null hypothesis pertains. With logistic noise, the observations are below this criterion value in every case. With Cauchy or with normal noise, the null hypothesis is clearly rejected at T=1/2. The case AB=11 is considered in the right-most column normal noise; and again the test statistic warrents rejection at T=1/2. These results might be summarized by saying that the asymptotic temperatures, calculated above for non-logistic noises, are reasonable approximations when they equal or exceed unit value.

Table 2. Divergence of the N-sample distribution from the theoretical distribution of the states of the four-unit NAND gate.

	input=(0,0)			(1,1)
T	LOGIS	CAUCHY	NORMAL	NORMAL
2.5	13.3	18.9	8.7	18.9
2.0	9.5	15.6	18.6	21.8
1.5	10.1	9.7	24.0	13.7
1.0	6.8	11.1	29.2	20.9
0.5	9.5	206.1	68.7	64.8

APPENDIX

The purpose of this Appendix is to derive the transition matrix of the Markov chain $\{\mathbf{s}_k, k=1,2,\ldots\}$. Let $\mathbf{s}$ and $\mathbf{s}^t$ be the network state as a column and row vector, respectively. Let $\mathbf{d}_j$ denote a column vector which has N components the i-th of which is δ_{ij} in terms of the Kronecker delta. Consider just the case of no input ($\mathbf{U} = \mathbf{0}$). Then $X_j = \mathbf{s}^t\mathbf{T}\mathbf{d}_j$ where $\mathbf{T}$ is the connectivity matrix subject to Hopfield's restrictions. The algorithm selects a j at random and computes $S_j = 1[X_j + Y]$, where $1[.]$ is the unit step and Y has d.f. F(y) and density f(y), which is symmetric about y=0.

We want the probability of a transition from state $\mathbf{s}$ to state $\mathbf{s} + \mathbf{ds}$, where $\mathbf{ds} = \mathbf{d}_j = \text{col}(\delta_{ij})$ if $s_j=0$ and $\mathbf{ds} = -\mathbf{d}_j = \text{col}(-\delta_{ij})$ if $s_j=1$. This probability, denoted $Q(\mathbf{s}+\mathbf{ds}|\mathbf{s})$, is proportional to 1/N, the probability that j is selected, and is given by

$$Q(\mathbf{s}+\mathbf{ds}|\mathbf{s}) = \begin{cases} (1/N)\Pr(Y+X_j > 0) & \text{if } \mathbf{ds}=\mathbf{d}_j \\ (1/N)\Pr(Y+X_j \le 0) & \text{if } \mathbf{ds}=-\mathbf{d}_j \end{cases}$$

in which X_j is determined by $\mathbf{s}$ as noted. These statements are the same as

$$Q(\mathbf{s}+\mathbf{ds}|\mathbf{s}) = \begin{cases} (1/N)F(\mathbf{s}^t\mathbf{T}\mathbf{d}_j) & \text{if } \mathbf{ds} = \mathbf{d}_j \\ (1/N)[1-F(\mathbf{s}^t\mathbf{T}\mathbf{d}_j)] & \text{if } \mathbf{ds}=-\mathbf{d}_j \end{cases}$$

because of the symmetry of the distribution of Y. For transitions of zero Hamming distance we shall have

$$Q(\mathbf{s}|\mathbf{s}) = 1 - \sum_{\mathbf{ds}} Q(\mathbf{s}+\mathbf{ds}|\mathbf{s}).$$

For transitions of distance more than one, the probability is zero, since the algorithm specifies that the interrogation of the units is one-at-a-time.

REFERENCES

1. D.H. Ackley et. al., A Learning Algorithm for Boltzmann Machines, **Cognitive Science**, 9(1985) 147-169.

2. R.H. Baran and J.P. Coughlin, Comments on a Population Model of China, **Automatica**, 22(1986) 255-256.

3. J.J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, **Proc. Nat. Acad. Sci. USA**, 79(1982) 2554-2558.

4. N. Metropolis et. al., Equation of State Calculations by Fast Computing Machines, **J. Chem. Phys.**, 21(1953) 1087-1092.

5. D.E. Rumelhart et. al., **Parallel Distributed Processing**: Explorations in the Microstructure of Cognition, vol. 1, Bradford, Cambridge, MA (1986).

6. D.L. Shaw and K.J. Roney, Analytic Solution of a Neural Network Theory Based on an Ising Spin System Analogy, **Physics Letters**, 74A(1979) 146-149.

7. D.W. Tank and J.J. Hopfield, Simple "Neural" Optimization Networks, **IEEE Trans. Circuits and Systems**, 33(1986) 533-541.

8. D.W. Tank and J.J. Hopfield, Collective Computation in Neuronlike Networks, **Scientific American**, 257(1988) 104-114.

Parallel Optimization via the Block Lanczos Method

Stephen G. Nash and Ariela Sofer
George Mason University

Abstract.

Traditional optimization algorithms are not easily adapted to parallel computers. Even though the linear algebra operations can be programmed in parallel, the costs associated with evaluating the objective function often overwhelm the linear algebra costs, and so some parallelism in the function evaluations is essential. This paper describes how such parallelism can be obtained by using the block Lanczos algorithm within a truncated-Newton method. This algorithm also admits parallelism in the linear algebra of the algorithm. The resulting algorithms are suitable for coarse-grained parallel computers. Details on arithmetic and communication costs are provided.

1. Introduction.

This paper describes an algorithm for solving

$$\text{minimize } f(x) \tag{1}$$

on a parallel computer. Here we assume that $f(x)$ is a smooth nonlinear real-valued function of n variables x.

A method for solving this problem was given in [10]. It is based on a truncated-Newton method [2]: given some initial guess x_0, at each iteration a search direction p is computed by approximately solving the Newton equations using a block Lanczos method; then a step is taken along that direction so that the function value decreases ($x_{k+1} = x_k + \alpha p$, where $f(x_{k+1}) < f(x_k)$).

As was shown in [10], such an approach can lead to a successful parallel algorithm. On a number of test problems, effective use of parallelism was made, both in the linear algebra operations, as well as in parallel function evaluations. The purpose of this paper is to analyze more carefully the algorithm used to compute the search direction, the block Lanczos method. We give here detailed information on the arithmetic and communication costs of that algorithm. Related discussions can be found in [13].

Here is an outline of the paper: In Section 2 we give a general discussion of the nonlinear optimization method. In Section 3, we show how parallel and vector computer hardware can be used within the block Lanczos method, and list its costs. Section 4 contains our conclusions.

Other approaches to parallelism in optimization algorithms are available; see, for example, [1] and [7].

2. The Optimization Algorithm.

The algorithm we have used to solve the problem (1) is a descent method based on a line search. If x_k is the current approximation to a solution x^*, then we set $x_{k+1} \equiv x_k + \alpha p$, where p is a local downhill (descent) direction for $f(x)$ at x_k, and $\alpha > 0$. The scalar parameter α is chosen so that $f(x_{k+1}) < f(x_k)$; techniques for computing α can be found in [4]. Under mild assumptions (see [3]) this algorithm can be shown to converge to a point where the gradient of $f(x)$ is zero, i.e., the first-order conditions for a minimum are satisfied. Our main interest here is the computation of the direction p, since this is typically the most expensive aspect of the optimization algorithm.

The classical approach to this problem is to use Newton's method. If we expand $f(x)$ in a Taylor series about x_k we obtain

$$\begin{aligned} f(x_k + p) &= f(x_k) + p^T g_k + \tfrac{1}{2} p^T G_k p + O(\|p\|^3) \\ &\approx f(x_k) + p^T g_k + \tfrac{1}{2} p^T G_k p \\ &\equiv f(x_k) + Q(p), \end{aligned}$$

where $g_k \equiv \nabla f(x_k)$ is the gradient of $f(x)$ at x_k, and $G_k \equiv \nabla^2 f(x_k)$ is the Hessian matrix. $Q(p)$ is a quadratic function in p, and it can be minimized by setting its gradient with respect to p equal to zero, resulting in a set of linear equations for p, called the Newton equations:

$$G_k p = -g_k. \tag{2}$$

If G_k is positive definite, then the solution of (2) corresponds to the minimum of $Q(p)$, and p is used as a search direction. Note that for this choice of p

$$f(x_k + \alpha p) \approx f(x_k) - \tfrac{1}{2}\alpha^2 g_k^T G_k g_k,$$

so that for small values of α we have $f(x_k + \alpha p) < f(x_k)$, whenever $g_k \neq 0$. Hence p is a local downhill direction unless the first-order optimality conditions are satisfied. If G_k is not positive definite, then a "nearby" positive-definite approximation to G_k should be used in place of G_k in (2) [5].

The resulting optimization method has an asymptotic quadratic rate of convergence, and this rapid convergence rate is enticing, but solving (2) can be expensive for large-scale problems, since it involves computing the matrix of second derivatives and solving a large system of linear equations at every iteration. As a result, we have chosen to use a different technique to compute a search direction.

Truncated-Newton methods are more suitable than Newton's method for the solution of large-scale optimization problems. The search direction p is computed as an approximate solution of (2), obtained using an iterative method for linear equations. Hence, a truncated-Newton method is a nested iterative method: there is an "outer" iteration for minimizing the function $f(x)$, and an "inner" iteration for solving the Newton equations (2). Here, in order to introduce parallelism into the algorithm, we

have chosen to use the block Lanczos method. A more common choice is the linear conjugate-gradient algorithm [8].

Truncated-Newton methods are attractive since they can be programmed to have low storage and arithmetic costs, not require the computation of the Hessian matrix, converge rapidly, and be applicable to large problems.

Earlier examples of truncated-Newton methods ([2], [9]), have been useful on vector computers [17], but have not offered much scope for exploiting parallel computers. By using a block Lanczos method for the inner iteration, parallel computations are introduced, where the degree of parallelism corresponds to the block size chosen, and hence can be adapted to the number of processors available. The block algorithms also retain a great many vector operations, and thus can be effective on parallel computers where each processor has vector hardware, such as the Alliant and Intel iPSC/2 machines.

Such block methods for solving linear equations have been described in [11] and [13]. The method used here is based on the block Lanczos method [16]; this is not the most straightforward choice, but it permits the numerically stable treatement of non-convex optimization problems (cf. [9]). (If the Hessian is not positive definite, the solution of the Newton equations may not be a descent direction; the Lanczos method allows the detection and correction of this difficulty. In addition, this approach is numerically stable for a non-positive-definite system of linear equations, unlike its theoretically equivalent partner, the linear conjugate gradient method.)

We now provide the formulas for the block Lanczos method. The algorithm minimizes $Q(p)$ as a function of p over a sequence of subspaces of increasing dimension. A more detailed discussion of the block Lanczos method can be found in the references cited above. The specific formulation given here is taken from [10].

Let G be an $n \times n$ symmetric matrix. The block Lanczos method with block-size m generates a sequence of $n \times m$ orthogonal matrices $\{V_i\}$ via:

> Pick V_1 so that $V_1^T V_1 = I_m$. Set $V_0 = 0_{n\times m}$, $\beta_1 = 0_{m\times m}$.
> For $i = 1, 2, \ldots$
> Set
>
> $$V_{i+1}\beta_{i+1} = GV_i - V_i\alpha_i - V_{i-1}\beta_i^T, \qquad (3)$$
>
> where $\alpha_i = V_i^T G V_i$ and the $m \times m$ matrix β_{i+1} is chosen so that $V_{i+1}^T V_{i+1} = I_m$.

V_i is computed as the result of a QR factorization applied to the columns of the right-hand side in (3). The matrix V_1 can be obtained using a random-number generator. We will assume that m divides n, although this is not necessary, and that the algorithm proceeds as above for the full n/m iterations (see below for a further discussion).

Define the block matrix $V_{(i)} = [V_1|V_2|\cdots|V_i]$; if exact arithmetic were used in the above algorithm, then we would have $V_{(i)}^T V_{(i)} = I$, and $V_{(i)}^T G V_{(i)} = T_{(i)}$ where $T_{(i)}$ is a block tridiagonal matrix with $m \times m$ blocks:

$$T_{(i)} = \begin{pmatrix} \alpha_1 & \beta_2^T & & & \\ \beta_2 & \alpha_2 & \cdot & 0 & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \beta_i^T \\ & 0 & & \beta_i & \alpha_i \end{pmatrix}.$$

A method for solving (2) is obtained as follows: let the first column of V_1 to be $g/\|g\|_2$, where g is the right-hand side in (2). Solve

$$T_{(i)}y_i = -V_{(i)}^T g = -\|g\|_2 e_1, \qquad e_1 = (1, 0, \ldots, 0)^T$$

for y_i. Then p_i, the i-th approximation to the solution of (2), is obtained from $p_i = V_{(i)}y_i$. This is equivalent to the block conjugate gradient method in [11]; both algorithms produce the same estimates of the solution of (2), if exact arithmetic is used.

This derivation is not suitable for computation since the resulting algorithm is not iterative. However, by adapting the derivation in [15], an iterative method can be developed. Assume now that G is positive definite; we will treat the indefinite case below. We use Gaussian elimination to factor the block tridiagonal matrix:

$$T_{(i)} = L_{(i)}D_{(i)}L_{(i)}^T, \qquad (4)$$

where $D_{(i)}$ is a block diagonal matrix whose blocks are themselves diagonal, and $L_{(i)}$ is a block lower bidiagonal matrix, with blocks the same size as in $T_{(i)}$. Define

$$U_{(i)} \equiv V_{(i)}L_{(i)}^{-T}, \qquad (5)$$

$$s_{(i)} \equiv -D_{(i)}^{-1}L_{(i)}^{-1}V_{(i)}^T g; \qquad (6)$$

both $U_{(i)}$ and $s_{(i)}$ can be generated iteratively. Then

$$p_i = V_{(i)}y_i = -(V_{(i)}L_{(i)}^{-T})(D_{(i)}^{-1}L_{(i)}^{-1}V_{(i)}^T g) = U_{(i)}s_{(i)}, \qquad (7)$$

and so an iterative algorithm, referred to here as the block Lanczos/CG method, is obtained. The formulas for the algorithm and their associated costs are described in more detail in the next section.

Minor adjustments to the algorithm are necessary if (a) the algorithm converges early, (b) m does not divide n, or (c) there is loss of orthogonality due to rounding errors. In such circumstances, when V_{i+1} is computed only the first $m_1 < m$ columns may be linearly independent. If this happens, then β_{i+1} will be an $m \times m_1$ matrix and V_{i+1} will be an $n \times m_1$ matrix. The remaining matrices in the algorithm will also have to be adjusted, but the formulas given above are still valid.

If $f(x)$ is not convex, then G may not be positive definite at every outer iteration. If this happens, then at some iteration i the LDL^T factorization of $T_{(i)}$ will not be numerically stable. Another factorization could be substituted (see [15] and [13]), but since we are more interested in obtaining a descent direction than in solving (2), alternative techniques may make more sense. It would be possible to use a modified matrix factorization,

as described in [4] and [9], or the algorithm could be stopped at the iteration where indefiniteness appears. Either approach will produce a descent direction.

3. Parallel and Vector Operations.

The block Lanczos method permits us to exploit parallel and vector capabilities in nearly every aspect of the computation of a search direction. In this section, we describe in detail one way of implementing the algorithm, the one used in [10], showing the arithmetic and communcation costs associated with each step of the algorithm. To simplify the discussion, we assume that the block size m is equal to the number of processors. This is not essential to the algorithm.

We shall consider each of the steps of the block Lanczos/CG algorithm in turn. We have implemented the algorithm on an Intel iPSC/2 which has no global memory; each processor has its own local memory. We tacitly assume that that n will be much larger than m, although the algorithm is valid without this assumption. Because of this, each processor stores only a small number of vectors of length n (one column of each of the $n \times m$ matrices V_i, V_{i-1}, GV_i, U_i, W_i, plus one work vector), but stores complete copies of the $m \times m$ matrices α, β, L_i, and $L_{i,i-1}$. If the number of processors were large, and hence m was large, then other approaches would be recommended; see the comments at the end of this section.

In the following discussion, we will number the processors from 1 to m, rather than the more usual 0 to $m-1$.

1. The Lanczos iteration—For nonlinear optimization, and particularly when the objective function $f(x)$ is expensive to evaluation, this will typically be the most expensive step in the method, and the place where effective use of parallelism will be most essential. This step involves m independent matrix-vector products, one for each column of V_i. If the Hessian G is available, GV_i can be computed using traditional techniques. However, more often a matrix-vector product will be approximated using [12]

$$Gv \approx \frac{g(x + hv) - g(x)}{h}$$

where $g(x)$ is the gradient and h is a finite-difference parameter. Since $g(x)$ is the right-hand side of (2), it is already available, and so a matrix-vector product can be approximated using a single gradient evaluation, and GV_i can be approximated by m independent gradient evaluations, one per processor. Thus we can make effective use of parallel gradient evaluations. If the gradient were not available, it could be approximated using a further level of finite differencing, without losing the parallelism discussed here.
   - Communication—This step requires that the gradient be sent to each processor (n real numbers).
   - Arithmetic (per processor)—One gradient evaluation, two vector additions ($2n$ operations), and two vector scalings ($2n$ operations).
2. Forming α_i and $V_i\alpha_i$—These matrices are computed simultaneously by sending the columns of the matrix V_i cyclically around the hypercube, considered as a ring. At the j-th step of this procedure, processor l computes $(\alpha_i)_{jl}$ where $j = [(l + j - 1) \bmod m] + 1$. Processor l then computes $(V_i)_j(\alpha_i)_{jl}$. Note that $(V_i\alpha_i)_l = \sum_{j=1}^{m}(V_i)_j(\alpha_i)_{jl}$.
   - Communication—Each processor sends/receives m vectors of size n.
   - Arithmetic (per processor)—Computation of the two matrices requires $2mn$ multiplications and $2(m-1)n$ additions.
3. Forming $V_{i-1}\beta_i^T$—This matrix is formed in the same way as $V_i\alpha_i$ above.
   - Communication—Each processor sends/receives m vectors of size n.
   - Arithmetic (per processor)—Forming the matrix requires jn multiplications and $(j-1)n$ additions on processor j. Since β_i is a triangular matrix, the arithmetic costs are slightly lower than before.
4. Forming the right-hand side in (3)—involves m independent vector additions.
   - Communication—None, after the previous steps have been completed.
   - Arithmetic (per processor)—2 vector additions ($2n$ operations).
5. Determining V_{i+1} and β_{i+1}—consists of a QR factorization of the right-hand side in (3), and can also be done in parallel [14]. A modified Gram-Schmidt algorithm is used [6].
   - Communication—Processor j sends one n-vector to $n-j$ processors, and receives $(j-1)$ n-vectors.
   - Arithmetic (per processor)—Forming of the factorization requires $2nj$ multiplications, $(2jn - j - n)$ additions, and one square root on processor j.
6. Factorization (4) of $T_{(i)}$—The matrix $L_{(i)}$ is block lower bi-diagonal with diagonal blocks L_i, and with subdiagonal blocks $L_{i,i-1}$. Let D_i (diagonal) be the i-th diagonal block of $D_{(i)}$. Then the factors of $T_{(i)}$ can be determined via

$$\begin{aligned} \alpha_1 &= L_1 D_1 L_1^T, \\ \beta_i &= L_{i,i-1} D_{i-1} L_{i-1}^T, \\ \alpha_i &= L_i D_i L_i^T + L_{i,i-1} D_{i-1} L_{i,i-1}^T, \quad i > 1. \end{aligned}$$

   These formulas correspond to LDL^T factorizations or back substitutions. These operations only involve $m \times m$ matrices. We have computed them on a single processor since m is small (at most 16) in our case. They can be performed simultaneously on all processors, as suggested in [13].
   - Communication—None.
   - Arithmetic (per processor)—Ignoring lower-order terms, formation of the new factors costs $m^3/2$ multiplications and additions.
7. Forming $U_{(i)}$ in (5)—Write $U_{(i)} = [U_1|U_2|\cdots|U_i]$ as was done with $V_{(i)}$. Then U_i can be computed by solving (via back substitution)

$$U_i L_i^T = V_i - U_{i-1} L_{i,i-1}^T, \quad U_0 = 0.$$

The second term on the right-hand side is formed in the same way as in step 3 above; combining the two terms involves m independent vector additions. To finish computing U_i requires repeated back substitution to solve for the *rows* of U_i.

- Communication—In forming the right-hand side, each processor sends/receives m vectors of size n. To solve for U_i, processor j sends n $(m-j)$-vectors and receives n $(m-j-1)$-vectors (except processor 1).
- Arithmetic (per processor)—To form the right-hand side requires jn multiplications and additions on processor j. Solving for U_i costs $n(m-j)$ multiplications and additions.

8. Forming $s_{(i)}$ in (6)—Divide up $s_{(i)}$ conformally to $U_{(i)}$: $s_{(i)}^T \equiv [s_1^T] \cdots [s_i^T]$. Then

$$s_1 = -D_1^{-1} L_1^{-1} V_1^T g;$$

note that $V_1^T g$ has only one non-zero component. At later iterations, we compute s_i from

$$L_i D_i s_i = -L_{i,i-1} D_{i-1} s_{i-1}.$$

These operations only involve vectors and matrices of order m, and we have chosen to do them simultaneously on all processors.

- Communication—None.
- Arithmetic (per processor)—This step requires $m^2 + 2m$ multiplications and $m(m-1)$ additions.

9. Forming p_i—Divide up $U_{(i)}$ and $s_{(i)}$ as above. Then (7) can be written in the form

$$p_i = p_{i-1} + U_i s_i,$$

and the right-hand side can be formed as the linear combination of m vectors, with in our case one on each processor. The Intel hypercube has a built-in operation of this type.

- Communication—Except for processor 1, each of the processors sends one n-vector. Processor 1 receives $m-1$ n-vectors.
- Arithmetic (per processor)—There are n multiplications per processor. Processor 1 performs mn additions.

10. Compute the residual $Gp_i + g$ (for the convergence test)—The formulas for the block Lanczos algorithm give

$$\begin{aligned} Gp_i &= GU_{(i)} s_{(i)} \\ &= (V_{(i)} T_{(i)} + [0\,0 \cdots W_i]) L_{(i)}^T s_{(i)} \end{aligned}$$

where $W_i = GV_i - V_i \alpha_i - V_{i-1} \beta_i^T$. Since

$$\begin{aligned} V_{(i)} T_{(i)} L_{(i)}^T s_{(i)} &= V_{(i)} T_{(i)} L_{(i)}^T D_{(i)}^{-1} L_{(i)}^{-1} V_{(i)}^T g \\ &= V_{(i)} V_{(i)}^T g \qquad g \end{aligned}$$

and

$$[0\,0 \cdots W_i] L_{(i)}^T s_{(i)} = W_i L_i^T s_i,$$

we obtain $Gp_i + g = W_i (L_i^T s_i)$. The term in parentheses is computed simultaneously on all processors, and the result is the linear combination of n-vectors, one per processor. The more obvious formula for calculating the residual was not used, to avoid an additional matrix-vector product. The resulting computations are almost the same as in the previous step.

- Communication—As in step 9.
- Arithmetic (per processor)—Each processor performs $n + m^2/2$ multiplications and $m^2/2$ additions. Processor 1 in addition performs $n(m-1)$ additions.

The above discussion shows that all the major steps (that is, all the $O(n)$ steps) in the block Lanczos/CG algorithm can exploit parallelism. In addition, many of these steps correspond to basic linear algebra subroutines (BLAS); for example, the inner products, linear combinations of vectors, and multiplications of vectors by scalars. These operations can be carried out using vector hardware or assembly-language instructions on many computers, in particular the Intel hypercubes and the Alliant. As a result, this algorithm should be well suited to parallel and parallel/vector computers.

The description above represents a column-wise organization of the algorithm. This is appropriate in this application because the matrix-vector products are produced one column per processor. Row-wise organizations are described in [13], where each processor stores a group of rows from each $n \times m$ matrix.

4. Conclusions.

We have presented a truncated-Newton method for minimization of a nonlinear function suitable for a parallel computer. It is based on a block Lanczos inner algorithm that can exploit parallel gradient evaluations. We believe that a successful parallel optimization algorithm for general use must be able to use parallel function/gradient evaluations, as this algorithm does. It should be especially useful when function/gradient evaluations are costly, and when the number of variables is larger than the number of processors available.

The algorithm is made up of steps that provide many opportunities for exploiting parallelism. The costs of these steps, both arithmetic and communication, have been described in detail. In addition, the lower level operations offer the possibility of further improvements in performance when the processors on the parallel computer in addition have vector capabilities.

5. Acknowledgments.

Stephen G. Nash is partially supported by Air Force Office of Scientific Research grant AFOSR-85-0222. Ariela Sofer is partially supported by National Science Foundation grant ECS-8709795 (co-funded by the U.S. Air Force Office of Scientific Research), and by Center for Innovative Technology grant CIT/SPC/87-005.

6. Bibliography.

[1] R.H. Byrd, R.B. Schnabel, and G.A. Shultz, *Parallel quasi-Newton methods for unconstrained optimization*, Tech. Report CU-CS-396-88, Dept. of Computer Science, University of Colorado at Boulder (1988).

[2] R.S. Dembo and T. Steihaug, *Truncated-Newton algorithms for large-scale unconstrained optimization*, Math. Prog. 26 (1983) pp. 190-212.

[3] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[4] P.E. Gill and W. Murray, *Safeguarded steplength algorithms for optimization using descent methods*, Report NAC 37, National Physical Laboratory (England) (1974).

[5] P.E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Prog. 28 (1974) pp. 311-350.

[6] G.H. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1983.

[7] S.P. Han, *Optimization by updated conjugate subspaces*, in *Numerical Analysis: Pitman Research Notes in Mathematics Series 140*, D.F. Griffiths and G.A. Watson, eds., Longman Scientific and Technical (Burnt Mill, England) (1986).

[8] M. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards 49 (1952) pp. 409-436.

[9] S.G. Nash, *Newton-like minimization via the Lanczos method*, SIAM J. Num. Anal 21 (1984) pp. 770-788.

[10] S.G. Nash and A. Sofer, *Block truncated-Newton methods for parallel optimization*, Technical Report 88-102, Dept. of Operations Research and Applied Statistics, George Mason Universtiy, Fairfax VA (1988).

[11] D.P. O'Leary, *The block conjugate-gradient algorithm and related methods*, Lin. Alg. Applics 29 (1980) pp. 293-322.

[12] D.P. O'Leary, *A discrete Newton algorithm for minimizing a function of many variables*, Math. Prog. 23 (1983) pp. 20-33.

[13] D.P. O'Leary, *Parallel implementation of the block conjugate gradient algorithm*, Parallel Computing 5 (1987) pp. 127-139.

[14] J.M. Ortega and R.G. Voigt, *Solution of partial differential equations on vector and parallel computers*, SIAM Rev. 27 (1985) pp. 149-240.

[15] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numerical Analysis 12 (1975) pp. 617-629.

[16] R. Underwood, *An iterative block Lanczos method for the solution of large sparse symmetric eigenproblems*, Computer Science Dept. Report STAN-CS-75-496, Stanford University (1975).

[17] S.A. Zenios and J.M. Mulvey, *Nonlinear network programming on vector supercomputers: a study on the Cray X-MP*, Operations Research 34 (1986) pp. 667-682.

A TOOL TO GENERATE FORTRAN PARALLEL CODE FOR THE INTEL IPSC/2 HYPERCUBE

C. Gonzalez, J. Chen, and J. Sarma., George Mason University

ABSTRACT

This paper reports on a software tool (pre-compiler) for translating sequential Fortran code to parallel form. We investigated and implemented a methodology for detecting data dependencies. A code generator was designed and implemented for the Intel IPSC/2 hypercube. This research concentrated on parallelizing do-loop structures, by dividing the data among the nodes. An outline and examples of the code generated for the cube manager and the nodes is presented. We discover that the use of this precompiler could potentially be an essential tool to use the hypercube effectively and efficiently.

Key Words: Pre-compiler, software tool, Fortran, hypercube, data dependence, code generation, supercomputer, parallelizing software.

1. INTRODUCTION

Modern supercomputers (i.e. parallel computers) provide hardware capabilites for parallel processing, but lack the software tools to support this parallelism. These supercomputer systems consist of a variety of multiprocessors, vectors-processors or multicomputers interconnected together in some fashion. Parallel computers are most effectively used when executing parallel object code. Unfortunately, most compilers for such systems can only process sequential source code. The parallelism is obtained by the use of explicit instructions inserted in the code. This restriction requires from the user to explore and detect the parallelism inside the problem and insert the commands for the concurrent programming [Seit-85]. This paper reports on the design and implementation of techniques for translating sequential code to parallel code. Our long term goal objectives is the construction of a tool that could convert valuable "old" sequential code to run on supercomputers.

Allen and Kennedy [Alle-82] preprocessed FORTRAN source code into FORTRAN 8x code in three steps: program normalization, dependence testing, and parallel code generation. This is the same general approach used in this research. Another related work is the family of vectorizers (KAPs) designed by Kuck and Associates, Inc which use loop interchanging [Davi-86, Huso-86, Mack-86]. An important difference with our work is that the KAPs' underlaying machines (ST-100, S-1, and Cyber 205) have a tightly coupled architecture, and our work was done for an Intel IPSC/2 hypercube, which is a loosely coupled architecture. Padua [Padu-86] made a comprehensive discussion on two types of parallel codes for compiler optimization: vector and concurrent. We Combined the above techniques, adding some source code optimization in front of the compiler. Our precompiler assumes "error-free" FORTRAN programs as input, and proceeds to parallelize the data for the do loops (SIMD model), but not the code.

This research was supported by the Center for Inovative Technology, contra.t No. SPC-87-005, and by the Army Research Office, contract DAAL03-87-K-0087.

2. SYSTEM MODEL

The functional decomposition of the model used for translating sequential code to executable code has five modules: the lexical analyser, the data dependence detector, the parallel code generator, the vectorizer, and the compiler. The produced final object code is composed of two different sets of code; one to be executed in the cube manager, and the other to be executed in each node of the hypercube.

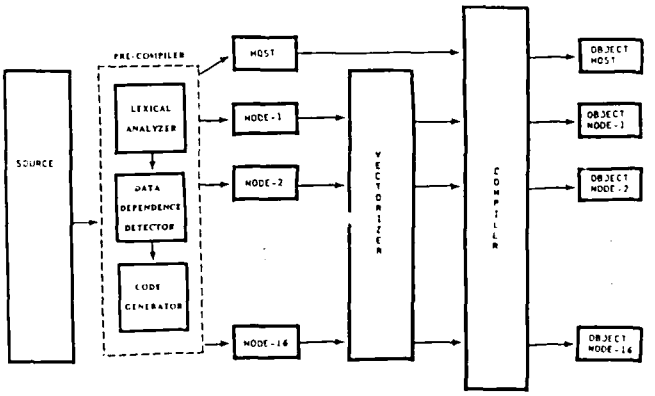


Figure.1 System Model

In this research we design and implemented the first three modules of the model described above (i.e. lexical analizer, data detector and code generator). We used the Fortran compiler from Green Hill Software Inc., for generating executable object code. We did not used the available vectorizer software and hardware in this project.

2.1 The Lexical Analyzer

The lexical analyzer translates Fortran source code into a sequence of tokens, fills in a symbol

table, and an array description table. A BNF of the simplified grammar subset of Fortran that we used, is presented in (Fig.2).

```
<prog>   ::= PROGRAM <id> ( <arredec> )n
                <statement> STOP END
<arredec> ::= DIMENSION <id> <index> ( , <id> <index> )
<index>  ::= ( <integer> ( , <integer> )n )  (where n=2)
<id>     ::= <letter> ( <letter> | <digit> )
<letter>    ::= A | ... | Z
<integer>   ::= <digit> ( <digit> )n
<digit>  ::= 0 | ... | 9
<statement> ::= ( <dostatement> | <simplestatement> )n
<dostatement> ::= DO <label> <id> = <doindex> , <doindex>
     ( , <doindex> ) ( <statement> )n <label> CONTINUE
<doindex> ::= <id> | <integer>
<label>  ::= <digit> ( <digit> )n        (where n=4)
<simplestatement>  ::=  A string of characters not
                        having a DO as first characters.
```

Figure.2 Subset of FORTRAN Grammar

2.2 Data Dependence Detector (DDD)

The DDD performs semantic analysis of the code to check for parallel do-loops. The input for this module are the tokens, symbol table, and array dependence table generated by the Lexical Analyzer. The semantic information is stored in a dependence table. This information includes: the line number where the read-, write-, and do-statements were found; also included is information about the variables and arrays on the left and right hand side of the corresponding statements. The DDD also outputs the array indexes and loop control variables, which are especially important for multi-dimension and multi-level do-statements.

2.3 Code Generator

The code generator makes use of information generated by the Lexical Analyzer, and the data dependence tables generated by the DDD (see figure 3.). If the current line generated is a no-parallelizable statement (i.e. with not data dependence implications), the code generator simply gets the information directly from the lexical analyzer output (step 1). If the current statement analyzed is a read-, write-, or do-statement, the code generator uses information from both the lexical analyzer and the depedence detector (steps 1 and 2). The next step synthesizes the information and writes it to a buffer. A final step produces the source for the host (file name: host.f) and the source for the node (file name: node.f).

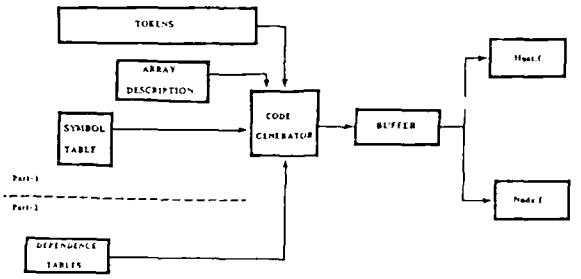
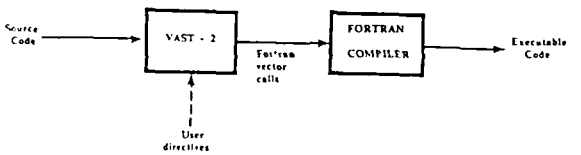


Figure.3 Code Generator and Tables

2.4 Vectorizer

The propose of the vectorizer software is to generate code that will use the vectorizer hardware board. The code produced by the code generator could be the input for this software. Hence, it also does data dependence checking and modifies the code by adding vector calls. The vector calls are supported by vector library and a vector processor attached to each node. The available software vectorizer is VAST-2 which according to user directives, changes program to expose array operation (Fig.4).




```
   DO 20 I=1,N          DO 20 I=1,N
   S=0.0                Y(I)=DDOT(N,A(I,1),LNA,X(1),1)
   DO 10 J=1,N       20 CONTINUE
   S=S+A(I,J)*X(J)
10 CONTINUE
   Y(I)=S
20 CONTINUE
```

source code *vector call output*

Figure.4 VAST-2 Program Development Sequence

2.5 FORTRAN Compiler

The iPSC/2 FORTRAN compiler used for this project was from Green Hills Software, Inc. The files host.f and node.f were compiled and linked to produce files: host and node, which are executable code.

3. DETECTION OF DATA DEPENDENCIES

The data dependencies among the statements were the deciding factors whether the "do loop" could be processed in parallel or not.

3.1 Basic Assumptions

We made certain assumptions in order to implement the data dependence analyzer. These assumptions were necessary so that we could handle simple loops before we added more complexities to it. The assumptions made were as follows:

1) There was only one level of do loops, i.e. no nesting of do loops was considered.
2) There were no equivalence statements in the source program.
3) The do loop was a very simple one (i.e. only arithmetic operations were performed inside the loop). There were no logical statements inside the loop (i.e. no transfer of flow statements), for which more complex analysis would be required.
4) The array indices were not greater than two. The DDD can be easily extended to include indices greater than two, without much problem.

3.2 Types of Dependencies

Data dependence relations between two statements determine if they can be executed in parallel. There are different types of dependencies between statements [Padu-86].

a) **Flow dependence**: can exist between two statements S1 and S2, if the data value in S1 is used in S2. Since statement S2 needs the value from S1, it cannot be executed unless statement S1 has finished executing. The following statements are an example of this type of dependence.

```
S1 :  A(I) = B(I) + C(I)
S2 :  D(I) = A(I) * 3
```

b) **Antidependence**: exists between two statements S1 and S2, if S1 uses a variable which is assigned a new value in statement S2. The following statements are an example of this type of dependence.

```
S1 :  A(I) = B(I) + C(I)
S2 :  B(I) = D(I) * 3
```

As can be seen from this example the two statements S1 and S2 cannot be executed in parallel as S1 uses the old value of B(I) which is later assigned a new value in S2.

c) **Output dependence**: between two statements can exist if a variable which is assigned a value in one statement and is later assigned a new value in another statement. The following statements are an example of this type of dependence.

```
S1 : A(I) = B(I) + C(I)
S2 : D(I) = A(I) * 3
S3 : A(I) = E(I) + F(I)
```

Statement S1 will contain a wrong value in A(I) if it is executed after statement S3. These statements have to be executed in the sequence they appear so that all the left hand side variables contain the correct value.

d) **Control dependence**: is the dependence which occurs from an "if" statement to the statements which are within the "if" statement block.

In the implementation of the precompiler, we considered only the first three types of dependencies. Control dependency was not analyzed because of the assumption that there were no logical statements inside the do loop.

3.3 Direction of the Dependencies

The direction of the data dependence relations also has to be analyzed inside the do loop. The data dependencies inside the do loop is found by analyzing the arrays and their subscripts. The following are the types of data dependence direction:

a) **Equal flow dependence**:

```
        DO  100 I = 1, K
S1 :    A(I) = B(I) + C(I)
S2 :    D(I) = A(I) * 3
100  CONTINUE
```

There is flow dependence between statements S1 and S2, but this dependence relation stays within the same iteration of the do loop. By which we mean that for any iteration, the value assigned to A(I) in statement S1 is used by statement S2 in the same iteration. Therefore we can say that there exists equal flow dependence between S1 and S2.

b) **Less than flow dependence**:

```
        DO 100 I = 2, K
S1 :    A(I) = B(I) + C(I)
S2 :    D(I) = A(I-1) * 3
100  CONTINUE
```

Statement S2 uses a value of the array variable A which was assigned during the previous iteration of the do loop, i.e. it uses an old value of the array variable A.

The flow dependence does not stay within the same iteration instead it flows from iteration i-1 to iteration i.

c) **Less than antidependence:**

```
        DO 100 I = 1, K-1
S1 :    A(I) = B(I) + C(I)
S2 :    D(I) = A(I+1) * 3
100  CONTINUE
```

Here statement S2 uses an old value of the array variable A which is assigned a new value in the next iteration by statement S1. Since S2 uses an old value there exists antidependence relation between the two statements S1 and S2. The dependency flow is from iteration i to iteration i+1.

The DO loop is parallelizable only if the statements inside the do loop block have an equal flow dependence relation.

3.4 Semantic Information

The data dependency analyzer generates the information needed by the code generator. The information required was the line number of the "do loop". Special treatment was required if there were any "read", "write", or "print" statements in the source program. These statements had to be processed by the cube manager, because the nodes can not access files. The dependency table was implemented by using arrays. The four statements: 'read', 'write', 'print', and 'do loop', are assigned integer values (0 to 3) and this is stored in the data dependency table, along with the line number of the statement in the source program. In addition information about the variables on the right hand side and left hand side of the assignment statement are also stored in arrays.

4. CODE GENERATION

The code generator produces parallel do loops for the nodes. The following sections describe some key issues, such as the format used by our model, communication overhead, buffer size, and work load for each node.

4.1 Communication Between Host and Nodes

A typical iPSC/2 hardware configuration is shown in Fig.5. The SRM functions are: support for program development, cube management, I/O interface, and gateway to host machines. The SRM hardware consists of a processor mother board (16 MHz 80386, 80387 coprocessor, and console terminal port), 8 Mbytes of 32-bit RAM, a DCM board, and an Ethernet TCP/IP communications board. Each node has a pair of 16 MHz 80386 and 80387 coprocessors, 1-8 Mbytes RAM, and a DCM board. The nodes communicate through message passing. The topology of the network is a hypercube. The cube we worked on has 16 nodes numbered 0 to 15.

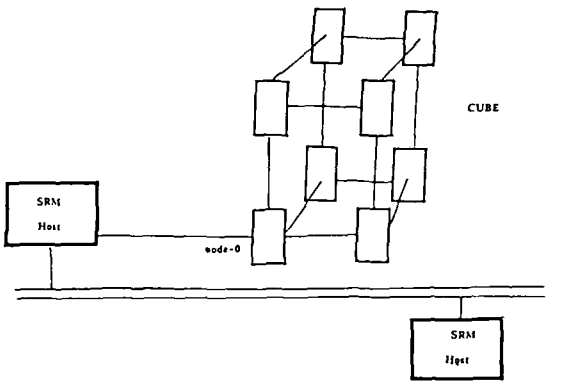


Figure 5. ISPC/2 configuration.

Communication Routines:

Our model uses the routines *csend* and *crecv* to communicate between the host and the nodes.

a) **csend(MSGTYPE, BUF, MSGLEN, NODEID, NODEPID)**

Sends a message between the nodes and the host, and waits until the whole message goes out.

-- MSGTYPE is the type of message. Used as message identifier. -- BUF is a one dimension array of integers or reals, containing the message sent out.
-- MSGLEN is the number of bytes in BUF (from 1 to MSGLEN) that will be sent out.
-- NODEID is the destination node/host id.
-- NODEPID is the process id at the destination node/host.

It is important to point out that the network path for communication is handled at the operating system level, thus, hidden at the FORTRAN level. Because of the hypercube topology we know the paths followed by each message, and we could use this information to minimize communication delays.

b)**crecv(MSGTYPE, BUF, MSGLEN)**

Receives the message from other nodes or from the host, and waits until the whole message is received.

-- MSGTYPE is the type of message, used as message identifier. If the MSGTYPE matches with that of the csend, the message arrives

at its destination.

-- BUF is one dimension array of integer or real, containing the message received.

-- MSGLEN is the upper bound number of bytes in BUF (from 1 to MSGLEN) will be received.

Following is an example for sending the first 100 real elements of an array from the host to all the nodes.

```
host:   REAL*4 BUFFOUT(2000), A(1000)
        INTEGER*4 TYPEOUT, ALLNODES, NPID, LENOUT
        DATA ALLNODES /-1/, NPID /1/, TYPEOUT /1/
        LENOUT = 400
        DO 301 I = 1, 100
        BUFFOUT(I) = A(I)
301     CONTINUE
        CALL CSEND(TYPEOUT, BUFFOUT, LENOUT,
                   ALLNODES, NPID)
        .
        .
        END

node:   REAL*4 BUFFIN(2000), A(1000)
        INTEGER*4 TYPEIN, HOST, LENIN
        DATA TYPEIN /1/, NPID /1/
        HOST = MYHOST()
        LENIN = 4000
        CALL CRECV(TYPEIN, BUFFIN, LENIN)
        DO 301 I = 1, 100
        A(I) = BUFFIN(I)
301     CONTINUE
        .
        .
        END
```

To reduce communication time, we made the message transferred as long as possible, instead of passing several short messages.

4.2 Disk I/O

In the iPSC/2 only the host can do a read or a write to disk. For a read-statement in the source code, we generate the following code (Fig.7). For a write-statement in the source code, we just simply copy the statement to the host.

4.3 The Host

The host manages the computations of each node, handles I/O, and communicates with other hosts. Our host code has features for supporting the above functions. For example, we set the do-loop control variables for handling workload of each node at run time. The host sends messages to all nodes concurrently, and waits to receive the results from all the nodes (Fig.8). A set of read statements in the source code will generate the

```
source: READ(1, 110) N1
        READ(1, 111) (A(I), I = 1, N1)
        .
        .
        END

host:   READ(1, 110) N1
        READ(1, 111) (A(I), I = 1, N1)
        LENOUT = N1 * 4 + 1 * 4
        BUFFOUT(1) = N1
        DO 301 I = 1, N1
        BUFFOUT(I) = A(I)
301     CONTINUE
        CALL CSEND(TYPEOUT, BUFFOUT, LENOUT,
                   ALLNODES, NPID)
        .
        .
        END

node:   LENIN = 2000 * 4 + 1 * 4
        CALL CRECV(TYPEIN, BUFFIN, LENIN)
        N1 = BUFFIN(1)
        DO 301 I = 2, N1 + 1
        A(I) = BUFFIN(i)
301     CONTINUE
        .
        .
        END
```

Figure.7 Read-Statements

code described in figure 7. This code will be inserted whenever the read statement is recognized by the code generator.

4.4 The Nodes

All nodes will run concurrently the same copy of the node program, but they will execute on different data (SIMD model). We use message passing between host and nodes, but not node to node. The message passing routines, *csend* and *crecv* were used to synchronizes the operations between host and nodes. Each node receives all the data. The data was not partitioned (i.e., all nodes get all the data) in order to maintain the simplicity of the code generated. Part of the continuation of this project will be the analysis of sending to every node only the data it will need to perform its computation.

Each node receives from the host the right hand side values of statements inside the do loop, and the do loop control values. It then calculates its own "ceiling", or upper bound of iterations for the loop. Then, calculates its inloop (initial value of the loop) and endloop (final value of the loop) values. Hence, different nodes will have different endloop and

inloop values. After each do loop is completed in the node, its results (i.e. the left hand side values) and the loop control values are sent back to the host. When the host receives these values, proceeds to load the values into its corresponding destinations.

```
PROGRAM header.
buffer declaration.
normal array and variable declaration.
special constant declaration.
special variable declaration.
equivalence-statements for control variables
                                and buffers.
data-statement for initializing special constants.
CALL SETPID(HOSTPID).
NNODES = NUMNODES().
CALL LOAD ('node', ALLNODES, NODEPID).
OPEN data files.
assign control variables.
compute message length (LENOUT) from control variables.
load output buffer (BUFFOUT) with data from right
                        hand side of do-loop.
CALL CSEND(TYPEOUT, BUFFOUT, LENOUT,
                        ALLNODES, NODEPID).
compute upper bound length (LENIN) of incoming
                        message from each node.
  DO 768 INODE = 1, NNODES.
  CALL CRECV(TYPEIN, BUFFIN, LENIN).
  move message to destination array.
768 CONTINUE.
rest of the code.
CALL KILLCUBE(ALLNODES, NODEPID).
CLOSE data files.
STOP.
END.
```

Figure.8 Host Program Outline

5. IMPLEMENTATION

The precompiler was developed using FORTRAN/VMS/VAX 8800. Then it was ported to the IPSC/2. This was done because compared with the VAX 8800, SRM is single user, and slower for program development. A complete set of examples and the source code for the Lexical Analizer, DDD, and Code Generator can be found in [Gonz-88].

6. CONCLUSIONS AND FUTURE RESEARCH

Many programs have been written in sequential FORTRAN. A precompiler that generates source code in parallel form can re-use most of the "old" FORTRAN programs to run on a supercomputer without redesigning and rewriting them. Some key factors which complicates this project are the number of nested do loop levels and any operation done with the indexes of the arrays.

We are working on a ***tutorial aid*** for directing FORTRAN programmer while using our FORTRAN pre-compiler to generate concurrent program. We plan to work on a model that supports **stochastic loop assignment**. This model will require flexible formats, and node to node communication. Another future research will consider **complex statements**, such as equivalence- and if-statements. A final future research is the implementation of the MIMD model in which **the program is divided** into segments (either subroutines or functions), download each of them to different node, and run them with their different data. We will include in all of our future models **performance measurement** and comparison with other models (hardware and software).

7. REFERENCES

[**Alle-82**] Allen J.R., and Kennedy K., "PFC: A Program to Convert Fortran to Parallel Form", Proceedings of the IBM Conf. on Parallel Computers and Scientific Computers, 1982.

[**Davi-86**] Davies J., et al., "The KAP/S-1: An Advanced Source-to-Source Vectorizer for the S-1 Mark IIa Supercomputer", Proc. of the 1986 International Conf. Parallel Processing, Aug-86.

[**Gonz-88**] Gonzalez C., Chen J., and Sarma J., "Experimental Results of Using a Precompiler to Parallelize Fortran Code," Technical Report #25. Center for Computational Statistics and Probability, George Mason University, May 1988.

[**Huso-86**] Huson C., et al., "The KAP/205: An Advanced Source-to-Source Vectorizer for the Cyber 205 Supercomputer", Proc. of the 1986 International Conf. Parallel Processing, Aug-86.

[**Mack-86**] Macke T., et al., "The KAP/ST-100: A Fortran Translator for the ST-100 Attached Processor", Proc. of the 1986 International Conf. on Parallel Processing, Aug., 1986.

[**Padu-86**] Padua D.A., and Wolfe M.J., "Advanced Computer Optimizations for Supercomputers", Communications of the ACM, Dec. 1986.

[**Seit-85**] Seitz C.L., "The Cosmic Cube", Comm. of the ACM, Jan. 1985, Vol.28, No. 1

[**Wolf-86**] Wolfe M., "Advanced Loop Interchanging", Proc. of the 1986 International Conf. on Parallel Processing, Aug., 1986.

MULTIPLY TWISTED N-CUBES FOR PARALLEL COMPUTING

T.-H. Shiau, Paul Blackwell and Kemal Efe, University of Missouri-Columbia

Abstract: It is known that by twisting one pair of edges of the N dimensional cube, the resulting graph denoted by TQ(N) has diameter N-1 instead of N. In this work, we show that by twisting multiple pairs of edges as well as pairs of buses (a bus is defined as a set of edges with certain common properties), the diameter becomes $\lceil 2N/3 \rceil$. The resulting multiply twisted N-cube, denoted by MTQ(N), preserves most of the desirable topological properties of the ordinary N-cube for parallel computing. A simple routing method is presented which can easily be implemented. Finally we discuss generalizations of MTQ(N) for which the diameters can be made even smaller at the expense of more complicated routing. The smallest diameter which can be achieved by this approach is $\lceil (N+1)/2 \rceil$.

KEY WORDS: Interconnection networks, Hypercube, Parallel processing

1. INTRODUCTION

An n-dimensional hypercube Q(n) = (V,E) is the graph with $N=2^n$ nodes each of which can be labeled by a unique n-bit binary number such that two nodes are adjacent if and only if their labels differ in exactly one bit position. The graph Q(3) is depicted in Figure 1.

Many multiprocessor computer systems use the hypercube as the interconnection network, i.e. each node of Q(n) is a processing element, usually with local memory, and the edges of Q(n) are the physical communication links. For example, the Cosmic Cube in Seitz (1985), iPSC of Intel Corporation (1985), NCUBE/10 of NCUBE Corporation (1986) and the Connection Machine in Hillis (1985) are all hypercube parallel computers, although the scales and granularities of parallelism of those computers very widely.

The popularity of the hypercube for interconnection networks stems from many of its nice topological properties. To name a few, the graph is *regular*, that is, each node has the same number n of adjacent nodes, has relatively small diameter which grows only logarithmically with respect to the total number of nodes, and has large minimum-bisection width (MBW) N/2. The MBW is the minimum number of edges which must be removed from the graph to separate it into two disconnected graphs with equal numbers of nodes (or different by 1 if the total number is odd). Small MBW implies severe limitations of parallel data routing between two parts of the system, while large diameter would mean large propagation delay in communication. Other properties of Q(n) can be found in Erdos and Spencer (1979), Folds (1977), Hart (1976), Mulder (1980), and Saad and Schultz (1985).

Although Q(n) has many desirable properties, it is shown in Esfahanian, Ni and Sagan (1987) that the diameter can be reduced by 1 by twisting any single pair of edges in any shortest cycle. For example Figure 2 shows the twisted cube with diameter 2. The twisted n-cube denoted by TQ(n), preserves most of

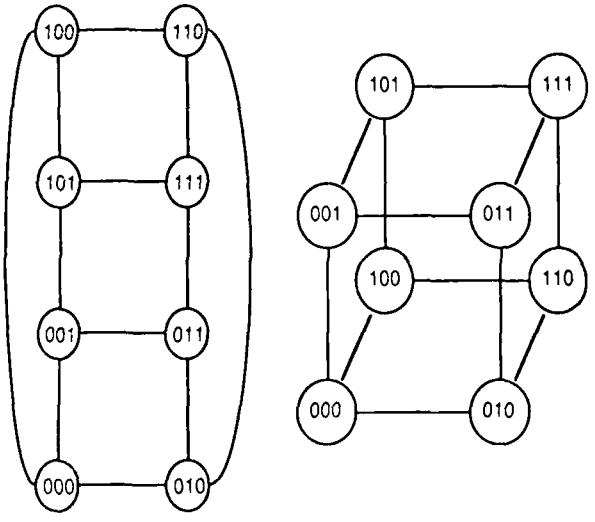


Figure 1. Q(3) drawn in two different ways

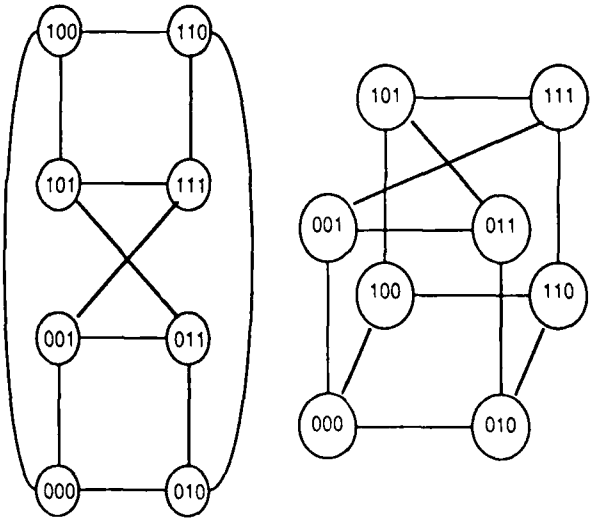


Figure 2. TQ(3) drawn in two different ways

This research is supported in part by AFOSR under Contract AFOSR-86-0124

the nice properties of Q(n). In addition, it contains the 2^n-1 node complete binary tree as a subgraph which is not a subgraph of Q(n). Independently, Blackwell et al. (1988) show that by properly twisting pairs of bundles of edges as a whole and pairs of edges within the bundles, the diameter can be reduced to $\lceil (n+1)/2 \rceil$ and most of those properties are still retained. This reduces by almost fifty percent the diameter of Q(n).

Although the fifty percent reduction of the diameter provides the potential for the same amount of reduction of the propogation delay of interprocessor communications, the routing is more complicated which would offset some of the advantages in practical applications. In this work, we show a much simpler way to construct a class of twisted hypercubes with diameter $\lceil 2n/3 \rceil$ for which a simple routing method exists.

In general, we can construct twisted cubes with diameter $\lceil (2+k)\, n / (3+2k) \rceil$. The greater the k, the more complicated the graph and the routing , and the closer the diameter is to the $\lceil (n+1)/2 \rceil$ of the graph in Blackwell et al. (1988)].

2. DEFINITION OF THE MULTIPLY TWISTED N-CUBES

Definition: A recursive definition is given as follows for multiply twisted n-cubes, MTQ(n), with diameter $\lceil 2n/3 \rceil$.

0. MTQ(0) = Q(0) which consists of a single node.

1. MTQ(3k+1) for k≥0, consists of two copies of MTQ(3k), G_0 and G_1, and 2^{3k} ($=|G_0|$) additional edges, called level 3k links, between G_0 and G_1 which defines an isomorphism f_{3k}: G_0 -> G_1 by $v_1 = f_{3k}(v_0)$ if and only if (v_0, v_1) is a level 3k link. In short, MTQ(3k+1) is constructed by linking two MTQ(3k) by a "straight bus" of 2^{3k} lines. The straight bus, in contrast to the twisted bus of Blackwell et al. (1988), makes the topology and routing very simple.

2. MTQ(3k+2) is similarly defined by two copies of MTQ(3k+1) and a straight bus of 2^{3k+1} level -(3k+1) links (edges).

3. MTQ(3(k+1)) consists of two copies of MTQ(3k+2) and a *twisted bus* of level 3k+2 links between them such that the eight copies of MTQ(3k) form a TQ(3), see Figure 3. More specifically MTQ(3k+3) = TQ(3) x MTQ(3k).

3. THE DIAMETER AND ROUTING

Theorem 1. The diameter of MTQ(n) is $\lceil 2n/3 \rceil$. The diameters of MTQ(3k+2) and MTQ(3k+3) are both 2k+2.

Proof: Induction on n.

Case 1 . n=3k+1. Let G_0 and G_1 be the two copies of MTQ(3k) in MTQ(n). Given any two nodes u,v, in MTQ(n), if they belong to the same copy of MTQ(3k), $d(u,v) \le \lceil (2/3)\, 3k \rceil = 2k$ by the induction hypothesis.

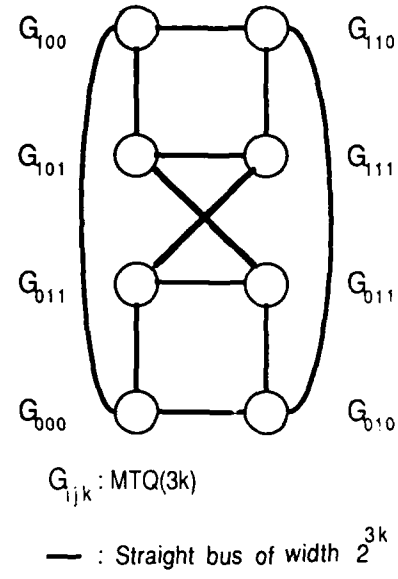


Figure 3 MTQ(3k+3)

Otherwise, assume $u \in G_0$, $v \in G_1$ and let $u' = f_{3k}(u) \in G_1$ where f_{3k} is the isomorphism. Then $d(u,v) \le d(u,u') + d(u',v) \le 1+2k$.

So the diameter of MTQ(n) ≤ $\lceil 2n/3 \rceil$. To show that equality holds, let again $u \in G_0$, $v \in G_1$, but also $d_G(u',v)=2k$. Let m=d(u,v), it suffices to show m=2k+1. Let $p=(u=s_0, s_1, \ldots, s_m=v)$ be any shortest path between them. There must be a level 3k link (s_i, s_{i+1}) for some i, 0≤i<m. By isomorphism

$$P'=(u'=s_0', s_1', \ldots, s_i'=s_{i+1}, s_{i+2}, \ldots, s_m=v),$$

where $s_j'=f_{3k}(s_j)$, is a shortest path with length m-1 between u' and v. So 2k=m-1.

Case 2 . n=3k+2. Similar to the previous case, we can show that diameter(MTQ(n)) = 2k+2 = $\lceil 2n/3 \rceil$.

Case 3 . n=3k+3. Because of the twisting of the two buses, we can go from any copy of MTQ(3k) to another by no more than two links. So diameter(MTQ(3k+3)) ≤ 2+diameter(MTQ(3k)) = 2 + 2k. By similar argument as in Case 1, the equality holds again. Q.E.D.

4. THE ROUTING METHOD

Routing on TQ(3) is straightforward since it consists of only eight nodes. The routing algorithm can either be directly hard-wired or by a lookup table of eight entries showing the outgoing link for each destination.The routing of MTQ(n) is simply a multi-level TQ(3) routing. Using the n-bit binary number labeling with the less significant bits for lower

level links, we can carry out the routing either bottom-up or top-down. By bottom up, we try to *correct* earlier the less significant bits, i.e. route the package to the intermediate node of which the label has the same less significant bits as that of the destination. Note that every three bits can be corrected in two steps by applying the same lookup table as that for TQ(3). So after $\lceil 2n/3 \rceil$ steps all the bits are correct. The top-down routing corrects the most significant bits first. The detail is omitted.

5. GENERALIZATION

Using basic modules other than TQ(3), we can construct different multiply twisted hypercubes with even smaller diameters. In Blackwell et al. (1988), a family of twisted cubes is given with diameter $\lceil (n+1)/2 \rceil$ where n is the dimension. For n=3, the graph is the same as TQ(n). To make this paper more self-contained, we shall describe the case for n=5 and thereby construct the multiply twisted hypercubes using it as the basic module.

We shall use the same notation TQ(n) for the new family of twisted cubes.

Definition:

(1) For n=2, TQ(n) is the same as Q(n).

(2) For n=3, TQ(n) is as in Figure 2.

(3) For n=4, TQ(4) is constructed from *four* copies of TQ(2), denoted by G_{00}, G_{01}, G_{10} and G_{11}, connected by four buses of width four as in Figure 4. Each bus is twisted so that the bus and the two copies of TQ(2) at its ends form a TQ(3).

(4) For n=5, TQ(5) is constructed from *eight* copies of TQ(2), $G_{b_2 b_1 b_0}$, b_i=0 or 1 for $0 \le i \le 2$, connected by (twelve) twisted buses of width four as shown in Figure 5. Again, each bus is twisted so that the two copies of TQ(2) and the bus form a TQ(3).

Remarks. The definition can be extended to arbitrarily large n. The resulting graph TQ(n) retains most of the nice topological properties such as regularity and strong connectivity, but with diameter

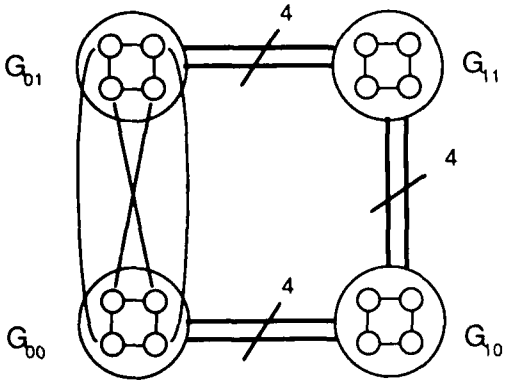


Figure 4 TQ(4) with diameter 3

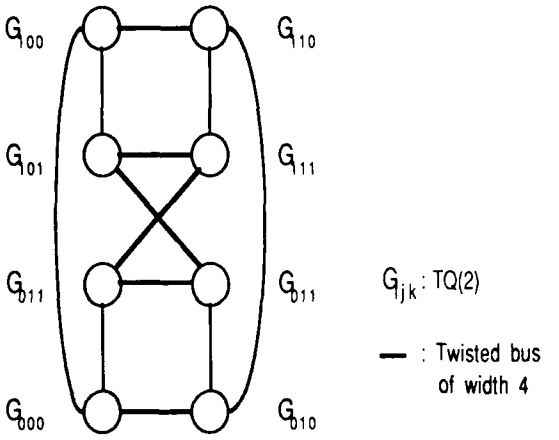


Figure 5 TQ(5) with diameter 3

only $\lceil (n+1)/2 \rceil$ and a more complicated routing algorithm. The detail is in Blackwell et al. (1988).

Using TQ(5) as the basic module, which has diameter 3, we can construct MTQ(n) similarly to Section II, so that diameter MTQ(n)= $\lceil 3n/5 \rceil$.

Formal definition is given as follows.

Definition .

0. MTQ(0) is a single node.
1. MTQ(5k+i)=TQ(i) x MTQ(5k), for i=1,2,...,5.

The routing is done by correcting 5 consecutive bits as a group by 3 links. The same routing algorithm for TQ(5) is used at each node after the active 5 bits are selected. Note that a lookup table for TQ(5) has 32 entries instead of 8 as in Section III, in which the diameter is larger.

By choosing yet bigger but more compact basic modules from Blackwell et al. (1988), we can define more compact MTQ with more complicated routing algorithms (or bigger lookup tables).

6. CONCLUSION

We show that by constructing the twisted cube hierarchically, one can reduce the diameter of Q(n) by a constant factor, such as 2/3 , and still keep the routing very simple. Theoretically, the constant factor can be made arbitrarily close to 1/2, although the additional complication of routing may make it undesirable.

It is interesting to note that in any hypercube machine such as The Connection Machine where the routing is done in parallel by correcting 1 bit at a time on the hypercube, if we reconnect the physical links to make it a MTQ(n) such that the bit positions which are corrected earlier correspond to higher level links, then the computer would work as usual without modifying the routing. The resulting routing algorithm would be able to route 2^n packages in parallel in n steps ,assuming no contention, on a twisted cube with diameter $\lceil 2n/3 \rceil$.

REFERENCES

Blackwell, P.K., Efe, K., Shiau, T.H., and Slough, W. (1988), "A Reduced Diameter Interconnection Network", Dept. of Computer Science, University of Missouri-Columbia, extended abstract submitted to the 2nd Symp. on the Frontiers of Massively Parallel Computation.

Erdos, P. and Spencer, J. (1979), "Evolution of the n-cube", *J. of Comp. and Math.with Appl.,*, 5, 33-39.

Esfahanian, A.-H., Ni, L.M. and Sagan, B.E. (1987), "The Twisted N-cube with Application to Multiprocessing", Dept. of Computer Science, Michigan State Univ., Dec. 1987.

Folds, S. (1977), "A characterization of hypercubes", *J. of Discrete Math.* ,17, 155-159.

Gustafson, J.L., Hawkinson S. and Scott, K. (1986), "The architecture of a homogeneous vector supercomputer", *Proc. of the IEEE 1986 Int'l Conf. Parallel Processing*, 649-652.

Harary, F. (1972): *Graph Theory*, Addison Wesley.

Hart, S. (1976), "A note on the edges of the n-cube", *J. of Discret Math.*, 14, 157-163.

Hillis, W.D. (1985): *The Connection Machine*, MIT Press, Cambridge, Mass..

Intel Corporation (1985): *iPSC System Overview.*

Mulder, M. (1980), "N-cube and median graphs", *J. of Graph Theory*, 4,107-110.

NCUBE Corporation (1986): *NCUBE Handbook*, Beaverton, Ore..

Saad, Y. and Schultz, M.H. (1985), "Topological properties of hypercubes", Tech. Rept., YALEU/DCS/RR-389, Dept. of Computer Science, Yale Univ., June 1985.

Seitz, C. (1985), "The Cosmic Cube", *Comm. ACM*, 28, no. 1, 22-33.

All-Subsets Regression on a Hypercube Multiprocessor

Peter Wollan, Michigan Technological University

Introduction. Parallel multiprocessor computers have been hailed as the next dramatic improvement in computing power. The object of this paper is to explore the use of one type of parallel computer (a distributed-memory system) in data analysis. All-subsets regression was chosen as a suitable vehicle with which to gain experience: it is a data-analysis procedure that is implemented in most statistical packages, yet it requires enough computation that standard mainframe computers may not provide enough power even for reasonably small problems; moreover, it appeared, in advance, to be inherently parallelizable.

Parallel computers come in essentially two varieties: shared memory, in which each processor has access to all of a common memory, and distributed memory, in which each processor has its own separate memory and sends messages to the other processors. In both cases, the goal has been to provide greater computational speed by dividing a problem into pieces which can be computed simultaneously, and then recombined into a solution. The shared memory systems have been found to be difficult to implement, both in hardware and system software, but once implemented they can be used, at least at some level, with comparative ease. Most users of Cray systems, for example, treat the system as if it were a single processor, and the effect of having several processors is to increase the number of users that can be serviced. Distributed memory systems, on the other hand, are comparatively easy and cheap to build, but require the user to explicitly parcel out computations to the separate processors, and to explicitly send messages among them to keep the computations coordinated.

The particular machine used here was an Intel iPSC-d4, which is a 16-processor hypercube. "Hypercube" refers to the communication links among the processors. Because of hardware limitations, it is not possible to connect every processor with every other. Several communication patterns have been tried, and have acquired names; the hypercube architecture seems to be the most common at this point. It can be thought of as placing 2^d processors at the vertices of a d-dimensional cube, with communication links provided only along the edges of the cube. Hence, each processor can directly communicate with d other processors, and messages to any other processor must be relayed. If the vertices are denoted by d-coordinate vectors of 0's and 1's, then a communication link exists between two vertices if the corresponding vectors differ in only one coordinate. Often, the vectors are thought of as d-bit binary integers, so that processor number 8 in a 4-dimensional hypercube, for example, is at vertex (1,0,0,0), and its neighbors, with which it can directly communicate, are processors 9, 10, 12, and 0.

The Intel iPSC has an additional processor, with its own memory, called the host. The host can communicate directly with all the other processors, which are called nodes. While communication between nodes is relatively fast, communication between host and nodes is relatively slow, and communication from the user to a node must pass through the host.

As noted above, it is necessary for the user to explicitly apportion computations among the nodes. Ideally, this will be done in such a way that all processors are kept busy the same amount of time, and so that no processor is forced to wait for another to complete an intermediate result. The procedure described here is not optimal, but is reasonably close, and uses the parallel nature of the machine in an acceptably efficient way.

Regressions were computed with the Sweep algorithm, which is well-known and widely used (see, for example, Weisberg (1987), p 60). Any implementation of the Sweep must use some method of checking for collinearity, if only to avoid dividing by zero. Largely out of curiosity, the method chosen was that proposed by Berk (1977); the behavior of this portion of the program turned out to be, in many ways, more interesting than the parallel part.

Section 2 describes the program, and gives some details about its components. Section 3 describes its performance, and includes a comparison with SAS Proc Rsquare. Section 4 concludes with some comments about using distributed memory computers for all-subsets regression, and some more tentative comments about using them for data analysis in general.

2. The Algorithm. All-subsets regression is often used to find the best set of predictor variables for a regression model. Given k predictors and a response, the linear regression is computed for each subset of predictors, and the best model is chosen by some

criterion. Since there are 2^k-1 non-trivial subsets, the number of models to be computed is very large even for reasonably small data sets.

The Sweep is an in-place matrix inversion algorithm; starting from a correlation matrix, it produces both standardized regression coefficients and $(1-R^2)$. It is attractive for regression for a number of reasons: it is easy to program, uses relatively little memory, and is numerically reasonably stable. It has two other features that are important for certain procedures such as stepwise regression and all-subsets regression: sweeping on a set of pivots produces the same result, no matter what order is used; and sweeping on a pivot a second time has the effect of deleting the corresponding variable from the model. Consequently, a predictor can be introduced into a model or deleted from a model in essentially the same amount of time. This feature allows the following approach to all-subsets regression: let the 2^k regression models correspond to vertices of a k-dimensional hypercube, where a model is described by a vector of 0's and 1's, with 1 in the ith coordinate indicating the ith predictor is present in the model. Then, the Sweep allows moving from one model to another along an edge of the hypercube, and a sequence of models determines a path along edges.

Using one processor, an efficient sequence of models corresponds to a path that passes through each vertex of the k-dimensional hypercube and does not pass through any vertex twice; if we add the requirement that the last model be one step away from the first (null) model, the path is a Hamiltonian circuit. There are many Hamiltonian circuits for the hypercube. One is given by the well-known Gray code (see, for example, Kohavi, 1978, p. 13), which allows computing the i^{th} vertex of the path from the binary representation of the integer i, using simple binary arithmetic.

Using 2^d processors, an optimal sequence of models corresponds to a set of 2^d paths, passing through every vertex, all starting at the origin (corresponding to the null model), which do not cross each other, and which are all nearly the same length (since they all start at the origin, they can't be exactly the same length). For certain k and d, such sets exist; however, there seems to be no way to extend solutions for small cubes to larger ones, and for some k and d there may be no solution.

However, a nearly optimal set of paths can be obtained from the Gray code mapping of the k-dimensional hypercube into the $2^d \times 2^{k-d}$ torus, as follows: a vertex of the cube is mapped onto a point in a $2^d \times 2^{k-d}$ rectangular grid. The row is determined by applying the Gray code to the first d coordinates, and the column by applying the Gray code to the remaining k-d coordinates. (The grid is a torus in the sense that the Gray code "wraps around" in both rows and columns) Each processor, then, is assigned a row of the grid. In order to get to the beginning of its row, the processor must introduce some variables, so there will be some duplication of effort among the processors; but no more than d variables need be introduced, so that the longest path is only d steps longer than the shortest.

The parallelization of all-subsets regression, then, can be described as follows: Each node is provided with the correlation matrix of the data. It introduces a set of predictors, to get to its row of the torus, then computes the models on its row, saving appropriate statistics from each model. When the node is done, it sends the collected statistics to the host for output. Communication among nodes is involved only at the beginning, when the data is being passed out, and at the end, when results are collected. The program uses recursive doubling to broadcast the correlation matrix to the nodes: the host sends the matrix to one node. The node sends the matrix to another; both nodes then send it to others, and so on, with the number of nodes receiving the matrix doubling at each step. This procedure uses both the inter-node communication links, which are very fast, and the parallel communication features of the hypercube architecture. Recursive halving could have been used to collect the results in one node for output, but this was found to be inefficient: the amount of output was fairly large, and there is a limit on the size of each message. Collecting all the output in one node, and sending it in smaller parcels to the host, was less efficient than simply having each node send its results to the host directly.

The computation of the regression models required checking for singularity of the matrix at each stage. Berk (1977) proposed a procedure in which the model is rejected (the predictor is not allowed to be introduced) if the trace of the submatrix corresponding to the model is greater than the tolerance divided by p, where p is the number of predictors introduced and the tolerance is chosen by the user (here, 1000). This was justified by an inequality involving the condition number. It is quite different from, and seems to be substantially more conservative than, the

procedures proposed by Stewart (1987) and Beaton, Rubin, and Barone (1976). Implementing Berk's procedure within the all-subsets regression program required some bookkeepping: it was necessary to keep track of the "official" model, given by the Gray code, and also the "actual" model, those predictors that were allowed to be introduced. Moreover, deleting variables often required recomputing the model from scratch, since a predictor that had been refused admittance earlier might be allowable in the smaller model.

The program was written in Intel's version of FORTRAN 77, which has a number of extensions to provide for communication between nodes. Each manufacturer has chosen its own set of extensions for this purpose, and Intel even changed the syntax substantially when it released the second version of the iPSC. Consequently, the code will not run on any other machine, and is not reproduced here; it is available from the author on request. The program's structure is as follows: the host program reads a correlation matrix from a file, sends it to one node, and then waits for output. As it receives output from each node, it writes it to a file. The same program executes on each node; the program can ask which node it is running on, and take different action depending on the answer. The node program begins by receiving the correlation matrix, then sends it on to others, using recursive doubling. The main program computes the variable to be introduced or deleted for the next model, using the Gray code algorithm; a subroutine computes the Sweep, and another subroutine recomputes the model when necessary. As models are computed, the program saves R^2 and two numbers describing which variables have been introduced; when all models have been computed, the collected results are sent to the host.

3. Results. The parallelization works well. The speedup factor is about .95 (that is, the time required for 1 processor, divided by the time required for n processors, is approximately .95n), for problems in the range of 8 to 15 predictors and up to 16 processors. In other words, a 16-processor machine takes slightly more than 1/16 of the time needed by a 1-processor machine for the same problem. This is not surprising, since it is generally communication overhead that matters as getting the output in readable form to the user, one could conclude that the Intel iPSC offers computing speed roughly comparable to an IBM mainframe, and at substantially lower cost. In fact, the time of 1.37 seconds is unfair to Intel: our particular machine has been running 20 to 30 times slower than than it should be, probably because of some undiscovered mis-specification in the instaliation of the operating system. In addition, the new version of the machine is a great deal faster.

However, SAS (and BMDP, and IMSL) use the Furnival-Wilson Branch and Bound algorithm for all-subsets regression (see, for example, Hocking 1976) which computes only the best models of each size. For the best 5 models of each size, for the 10-predictor problem described above, SAS Proc Rsquare required only .43 seconds.

Another feature of the program, Berk's singularity check, behaves in an interesting way: in effect, it gives a means of finding the best acceptable model. Even though the output, in its present form, displays only R^2 and two coded integers describing the models, one can easily scan the output and find those models which both have high R^2 and pass the tolerance test. For example, for the Longley data (see, for example, Beaton, Rubin and Barone, 1976) one can quickly see that the best acceptable three-variable model is obtained by fitting the variables Unemployment, Size of Armed Forces, and Year, where "best" is in the sense of greatest R^2, and "acceptable" is in the sense of passing Berk's tolerance test, with tolerance equal to 1000. One also sees that adding a fourth variable, Noninstitutional Population, yields a slightly higher R^2 and still passes the test. It should be noted that these models are quite different from the ones Beaton, Rubin, and Barone suggested; furthermore, it is difficult or impossible to obtain qualitatively similar results from SAS, BMDP, or IMSL. SAS Proc Rsquare apparently does not check for collinearity or tolerance in any way at all; IMSL subroutine RLEAP does check for singularity, but the manual does not describe what method is used; and BMDP-9R carries out a tolerance check, but terminates when a model fails the test.

4. Conclusions. All-subsets regression is a large enough computing problem for parallel computers to be potentially useful. However, the experience gained here indicates that distributed-memory systems, as they are presently designed, have serious shortcomings which make their use for this problem doubtful in spite of their speed.

The Furnival-Wilson algorithm is clearly the best way to screen a large number of models, which is generally what people want to do when they use an all-subsets regression program. The advantage of

computing only the good models is already substantial for 10 predictors, and it increases dramatically as the problem gets larger. It may be possible to code this algorithm for a distributed-memory system, but it is not at all clear how to do it. In fact, the algorithms that have been successfully parallelized for these systems have tended either to assign distinct, essentially independent computations to each processor, as was done here, or to implement large matrix methods and (roughly speaking) give a portion of the matrix to each processor. The Furnival-Wilson algorithm is of a different form altogether: its efficiency derives from eliminating potential cases, and at any given time there is not a great deal of computing to be done.

Computing every regression model, as is done here, is not likely ever to be very attractive. However, it does allow multiple, conflicting screening criteria. In particular, Berk's tolerance check is potentially interesting as a means of diagnosing and handling multicollinearity. It may be possible to include multiple criteria in the Furnival-Wilson algorithm; this would achieve the best of both worlds.

Regarding the general use of distributed-memory systems for data analysis, several limiting features have become apparent. First, input and output are severely restricted. As they are designed now, these machines are completely inappropriate for such input-bound problems as reduction of huge data sets. There is not only a bottleneck at the host, there is a limit on the size of messages sent between nodes; for the all-subsets regression program on the Intel iPSC, the nodes become unable to send their accumulated output in a single message when the number of predictors is only 15, even though the output consists of only three numbers per model. (The new version of the iPSC also has a limit, but somewhat larger). One could reasonably want to see a residual plot, or a set of several diagonostic plots, for each model; that is not practical now.

Second, the process of programming the system is very difficult. To some extent, this is due to the fact that the machines are new, and programming tools are still being developed. For example, the new version of the Intel iPSC comes with a debugging package that represents a major improvement. However, programming any distributed memory system requires using some elementary programming structures that are very different from those taught in traditional programming courses. One example is the Gray code, needed to describe which nodes are adjacent to which others; another is the recursive doubling communication algorithm. These, and others, are part of the basic language of the program, just like arrays and Do loops. In addition, parallel algorithms require a substantially different way of thinking about problems.

Accumulated experience will improve both the programming tools and the programmer's knowledge and skill, but there appears to be a fairly large class of problems that simply aren't suited for distributed-memory systems. One example seems to be the Furnival-Wilson algorithm. Another is the computation of a correlation matrix: covariances can be computed in parallel by giving each node a set of cases, and computing partial sums, first for means and then for cross products, and exchanging the partial sums among the nodes so that each node ends up with the full covariance matrix. However, the final step, going from covariances to correlations, is difficult to paralellize efficiently.

Distributed memory parallel computers are very fast and powerful; but programming them requires new techniques and unfamiliar tricks, and their full power may be usable only for certain kinds of problems. Overall, they appear to be special-purpose machines, whose capabilities satisfy only some of the needs of data analysis.

References

Beaton, A. E., Rubin, D. B., and Barone, J. L. (1976). The acceptability of regression solutions: another look at computational accuracy. Journal of the American Statistical Association, 71,158-168.

Berk, K. N. (1977). Tolerance and condition in regression equations. Journal of the American Statistical Association, 72, 863-866.

Hocking, R. R. (1976). The analysis and selection of variables in linear regression. Biometrics, 32, 1-50.

Kohavi, Z. (1978). Switching and Automata Theory, 2nd ed. McGraw-Hill, New York.

Stewart, G. W. (1987). Collinearity and least squares regression. Statistical Science, 2, 68-100.

Weisberg, S. (1985). Applied Linear Regression, 2nd ed. Wiley, New York.

Testing Parallel Random Number Generators

Mark J. Durst, Lawrence Livermore National Laboratory

As multiprocessor computers and networked computations become more common, there is a need for parallel pseudo-random number generation. This can be thought of as the provision of many streams of pseudo-random numbers, which should appear to be independent within each stream and across streams. Some ways of constructing tests for parallel random number generators are discussed, along with the computational limits on them. Experience using these tests to construct parallel random number generators for the Cray X-MP has failed to produce a particularly powerful set of tests, and so the importance of constructing computation-specific tests is stressed; some guidance is offered for these constructions.

1 Introduction

The use of multiprocessor computers and networks of computers to solve serious problems with parallel computations is becoming more common. Since these computers are in general asynchronous, standard system pseudo-random number generators (RNG's) are insufficient, as they lack *reproducibility*: a guarantee that different runs of a program will give the same result. Non-reproducible runs should only vary at the statistical level, and so production runs of a simulation or Monte Carlo calculation can often relinquish reproducibility and use standard system RNG's. However, for debugging purposes and for complex Monte Carlo calculations requiring intricate traces (for instance, where one wishes to target specific histories for future variance reduction), one must be able to reproduce computations exactly, and so parallel random number generators (PRNG's) are required.

A PRNG can be viewed as a method for producing multiple streams of pseudo-random numbers, and there is experience with such methods; some discussion occurs in Frederickson et al. (1984), and Schruben and Margolin (1978, p. 507) comment: "When two sets of pseudorandom number streams are generated using different randomly selected vectors of seeds, the two resulting time series samples... are typically observed to be uncorrelated." Past work has focused on providing a very small number of streams; current computing demands many more. Relatively inexpensive computers are now available with a thousand processors, and the ability to create logical tasks which do not necessarily correspond to physical processors creates programs with a need for even more streams; a production code at LLNL demands the availability of seventy million (short) streams.

Without *ad hoc* modifications (see Durst (1988)), the most promising techniques for parallel random number generation involve splitting up the cyclic stream of a given random number generator into substreams. This provides substreams of sufficient size for most current applications (particularly if one splits the stream from a generalized feedback-shift register or lagged-Fibonacci generator), but strains the discrepancies of current RNG's. Few applications of standard RNG's use the independence of more than about a dozen dimensions; good discrepancies at these dimensions are provided by the above generators, as well as by large-modulus (48 bits and above) congruentials. However, parallel computations may require that some dozens of streams appear independent in a dozen or so dimensions; the required discrepancies in hundreds of dimensions are far beyond the discrepancies of congruential RNG's (CRNG's), and are at or beyond those for generalized feedback register generators (GFSR's).

While conceding the theoretical shortcomings of current methods, though, it should be pointed out that many—perhaps most—Monte Carlo calculations and simulations do not have sophisticated independence requirements, and can even succeed by appropriately splitting a CRNG. Empirical tests should be used to verify minimally good properties of PRNG's, to provide simplified paradigms of complex calculations, and to check for the necessity and efficacy of modifications to PRNG methods. While such tests are well-known for standard RNG's (see, for exam-

ple, Knuth (1981) and Marsaglia (1985)) they generally focus on testing a single stream; here tests are required of the interdependence of many different streams (it is assumed that standard RNG tests will be used to check the quality of individual streams). In this paper a few basic ways of constructing tests are discussed, with some recommendations and comments on computational constraints.

2 Some PRNG Tests

Correlation tests are not very powerful, working only with pairwise behavior and detecting only the most serious dependencies. However, they are an important class of tests, since (as will be seen) no other testing regimen used here can verify much about correlations at many lags. The desire with correlation tests is to guarantee, for m streams, each of length n, that correlations of lag k or less are under control. For many applications, k can be on the order of one or two dozen, while sensitive applications may require that k be on the order of several hundred. There are omnibus tests for the independence of multivariate normals (see Anderson (1958), Chapter 9) which can be used asymptotically. One can also test the correlation coefficients with a Bonferroni test. If the correlations are computed directly, the computation time is $O(km^2n)$, but Fourier techniques can bring this down to $O(m^2 n \log(n))$. These tests are most effective at finding streams which are exact duplicates or antithetic variates at relatively small lags.

Latitudinal tests are a general way of constructing tests for PRNG's. Ordinary RNG tests split a sequence longitudinally, with a short sequence providing the numbers needed to compute one observation, and adjacent short sequences used to provide repeat observations. In a latitudinal test, one number from each of a fixed number of streams is used to provide one observation; repeated observations are then obtained by proceeding longitudinally. Of course, latitudinal and longitudinal testing can be combined. For instance, a four-dimensional equidistribution test can be used to compare two longitudinal dimensions of two streams. Latitudinal tests can be constructed from any test which always generates an observation from the same size set of numbers. It is not obvious how to adapt other tests, such as gap tests and runs tests, for latitudinal use. Tests for latitudinal use usually should not depend on the order in which the numbers appear; however, in some applications, there may be a natural stream ordering, which should be incorporated into tests. In small dimensions, equidistribution tests on unit hypercubes can be used. For somewhat larger dimensions, permutation and partition tests can detect some bad deficiencies. For still larger dimensions, most tests have been designed *ad hoc* and have not been very useful in testing PRNG's, but should probably be considered: examples are collisions tests and tests based on transforming the maximum of a number of uniforms to uniformity (Knuth (1981), pp. 68–70), and the "Birthday Spacings" test of Marsaglia (1985).

Given the insensitivity of high-dimensional latitudinal tests, one would ideally compute tests for all possible subsets in low dimensions. With m streams, a test of dimensionality j, and k lags under consideration, this would involve $2k\binom{m}{j}$ tests. This is infeasible unless either m is small (a dozen or two) or j is very small (2, 3, or 4). Since uniformity in the lowest dimensions is always desired, it is recommended to use an equidistribution test in dimension 2 (and 3 if feasible) on all pairs (triplets) of streams. For formal testing, Bonferroni tests should be used until information on the joint distribution of the $\binom{m}{j}$ possible p-values is available.

Another possibility is to compute a small random sample of the $\binom{m}{j}$ possible tests. If $m \gg j$, then all these tests should be effectively independent. Note, however, that the probabilistic guarantees afforded by such a test are only useful if streams with a small fraction of dependencies will result in a successful computation.

3 Experience

We have done empirical testing in the course of constructing three parallel random number generators:

- A default vectorized PRNG, intended for simple use with a moderate number of tasks (up to several hundred) on a Cray X-MP computer;
- A scalar PRNG for a physics simulation using up to many millions of short (at most several thousand) streams on a Cray X-MP computer, and
- A special-purpose PRNG for a physics computation on an eight-processor Alliant computer.

For the first two generators, we chose to split the sequence from the default Cray RNG RANF, a mul-

tiplicative congruential generator with modulus 2^{48} and multiplier 44485709377909. This choice was made for three reasons: compatibility with results from older codes; availability of the spectral test (which can be derived, for splitting use, from the work of Percus and Kalos (in press)); and, for the second generator, small state space (one word). For the first generator, which requires initialization to select a maximum possible number of streams, we considered forcing an odd number of streams and evenly (or nearly evenly) splitting the sequence, evenly splitting the sequence and then backing off a fixed amount, and evenly splitting the sequence and then backing off a fixed fraction. Testing was intended to compare these three schemes. For the second generator, where the number of streams was only bounded by 70,000,000, we decided to provide a default, even jump between streams, and so wanted to use testing to help select that jump. The third generator required that either very long streams or very many streams be available, which strained the congruential; we decided to split the sequence from a lagged-Fibonacci. In the absence of deterministic testing, we decided not to risk even sequence spacing, and so generated starting points with a congruential generator, in the hope of distributing starting points at random through the sequence of the lagged-Fibonacci. We used testing to check for overall interstream quality and to insure that the starting point mechanism was not too bad.

We also tested various straw men. One was an even split of the aforementioned RANF into 2^e streams for e from 1 up to 16. Another was a split which either used very small lags (we tested 1, 5, and 40) or used small lags to an even split as above. A final set of straw men were generators which generated duplicate streams (both a small number and a small fraction) and streams which were mixtures of other streams.

The tests used were correlation tests, a two-dimensional equidistribution test (five bits), a four-dimensional equidistribution test (three bits), permutation tests up to dimension six, a collisions test in dimension 20 (which tested only the top bit), and the Birthday Spacings test in dimension 256. The two- and four-dimensional tests were used on all subsets for up to 32 and 16 streams respectively, and all latitudinal tests were used in random combinations for 2^e streams with e up to 16. For the random combinations, some tests were done with a randomly chosen lag on each stream. Lags were chosen with a geometric distribution, with the probability of zero lag equal to 1/2.

The tests were very effective at discovering the straw men, with the exception of the Birthday Spacings test. The low-dimensional tests differed from the null hypotheses most spectacularly. Of course, the unlagged tests did not uncover the problem with the small-lag streams; those were most reliably detected by the correlations tests, which worked surprisingly well, even at detecting the straw men involving even splits. For even splits with e above 10, the strongest interstream dependencies involve only a small fraction of the streams; still, as long as several hundred randomly chosen tests were performed, the deficiencies were noticed.

The tests did not discover deficiencies in the schemes under serious consideration; whether this indicates lack of power in the tests or good parallel random number generators is unclear. We lacked exact joint distributions when testing all subsets, but no values ever exceeded the Bonferroni limits. Tests were iterated and analyzed as in Fishman and Moore (1982), but still no clear pattern of failure emerged. The testing did twice uncover what turned out to be programming errors which generated bad or badly dependent streams, so there may be some hope for these specific tests.

4 Recommendations

While congruential schemes have severely limited discrepancies (the same limits first described by Marsaglia (1968) apply), they survive tests like these. This indicates that passing such tests is a minimal requirement for parallel random number generators. Better tests in large dimensions remain of interest, as the power of existing tests in hundreds of dimensions leaves much to be desired.

For specific computations, three recommendations can be made. The first is that tests should be tailored to the application, as recommended by Marsaglia (1985). Poor results from bad ordinary RNG's may provide some guidance in finding specific tests to turn into latitudinal tests. The second recommendation is that specific streams with crucial independence requirements should be identified and tested heavily. For instance, if streams are used spatially, then each stream should be tested against all nearby streams. A final recommendation is that users of parallel random number generators should have available an *ad-hoc* scheme for improving PRNG's, using shuffling or

combination (see Durst (1988)). Suspicious simulation or Monte Carlo results can then be submitted to the improved scheme for validation. Of course, the improvement scheme should be submitted to testing to ensure that it at least does not degrade the PRNG.

Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

References

Anderson, T.W. (1958), *An Introduction to Multivariate Statistical Analysis*, New York: Wiley.

Durst, M.J. (1988), "Improving Parallel Random Number Generators" (in preparation).

Fishman, G.S., and Moore, L.J. (1982), "A Statistical Evaluation of Multiplicative Congruential Random Number Generators with Modulus $2^{31}-1$", *Journal of the American Statistical Association*, **77**, 129–136.

Frederickson, P.O., Hiromoto, R., Jordan, T.L., Smith, B., and Warnock, T. (1984), "Pseudo-Random Trees in Monte Carlo," *Parallel Computing*, **1**, 175–180.

Knuth, D.E. (1981), *The Art of Computer Programming*, vol. 2: *Seminumerical Algorithms*, 2nd edition, Reading, MA: Addison-Wesley, Chapter 3.

Marsaglia, G. (1968), "Random Numbers Fall Mainly in the Planes," *Proceedings of the National Academy of Sciences*, **61**, 25–28.

Marsaglia, G. (1985), "A Current View of Random Number Generators," in *Computer Science and Statistics: Proceedings of the Sixteenth Symposium on the Interface*, (L. Billard, ed.), New York: North-Holland.

Percus, O.E., and Kalos, M.H. (in press), "Random Number Generators for Ultracomputers," *Journal of Parallel and Distributed Computing*.

Schruben, L.W., and Margolin, B.H. (1978), "Pseudorandom Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments," *Journal of the American Statistical Association*, **73**, 504–520.

VII. DENSITY AND FUNCTION ESTIMATION

Interactive Smoothing Techniques

Wolfgang Härdle, Universität Bonn

Abstract

For effective implementation of smoothing techniques a *conditio sine qua non* is an interactive computing environment. We describe some of the logical structures that we find convenient for interactive smoothing. These structures are implemented in XploRe - a computing environment for parameter free regression and density smoothing in high and low dimensions.

0. The Smoothing Analysis Cycle

Smoothing means parameterfree estimation of regression and density curves. If $X \in \mathbb{R}^d, Y \in \mathbb{R}$ denote a pair of random variables, it is the task of regression smoothing to estimate the mean function $m(\cdot) = E(Y|X = \cdot)$ from an independent sample $\{(X_i, Y_i)\}_{i=1}^n$. Density smoothing consists of finding good approximations to the density function $f(\cdot)$ of X from an i.i.d. sample $\{X_i\}_{i=1}^n$. If no parametric restrictions are imposed on these curves the smoothing technique is nonparametric or parameterfree and is typically based on "pooling neighboring information", see Stone (1977).

There exists a wide variety of methods for parameterfree estimation, see e.g. Silverman (1986). These methods have more or less the same asymptotic sharpness but behave quite differently for finite sample size. This is a situation where the computer can be a very good assistant: smoothing means function estimation and therefore different results can only be studied in the form of comparing graphs or tables of values. Another scenario in this setting is to form residuals and to examine them in an iterative way for non-fitted or overfitted structure, see e.g. the backfitting procedure of Hastie and Tibshirani (1987). Here again the computer is agreat assistant in trying several alternatives.

Smoothing in dimensions of X bigger than two creates difficulties on the computational and on the statistical side. First of all one cannot study the full fit function without additional "artificial dimensions". Scott (1986) proposes to use time as this dimension and presents changing density contours for dimension $d = 4$. Secondly, in data sets with moderate sample size there is not enough data to perform the "local data pooling" in an effective way. (Theoretically speaking, this means that the rate of convergence of nonparametric smoothers is extremely slow for large dimensions d, see Ibragimov and Hasminski (1982) and Stone (1982).) *Additive models* reduce this dimensionality problem but require quite a bit of machine power e.g. the Projection Pursuit Regression (PPR) algorithm by Friedman and Stuetzle (1981). Interactive control of such an additive model comes into consideration, where one would like to see slightly different projections and corresponding alternative smooth fits in a small neighborhood of some currently favored fit.

Even if a single smoothing method is preferred the choice of smoothing parameter is rather delicate. A wide variety of algorithms yield (asymptotically) "optimal curves" but these can be quite different for finite sample size, see Marron (1986).

Summarizing the above situations we can state that the applied scientist will experiment with different smooth fits and try several alternatives in an iterative way. The typical scenario might be described as follows. The scientist starts with some initial *smooth* curve and then *examines* the graph and perhaps residuals. In a further step he *evaluates* this information perhaps using prior information on forms or structure of the current curve, then he may want to *compare* this current curve with an alternative. This iteration procedure can be called a smoothing analysis cycle as depicted in Figure 0.1.

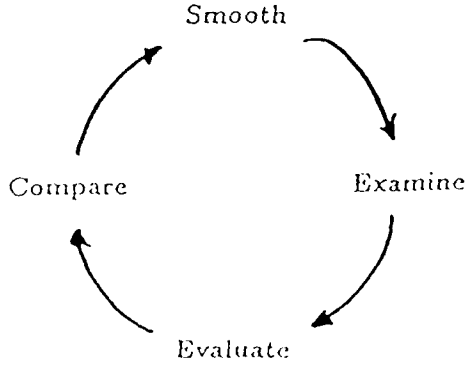


Figure 0.1. The smoothing analysis cycle

This cycle might be performed several times in an improvisional way before one or several satisfactory results are obtained (McDonald and Pederson, 1986). It is obvious that one needs a highly interactive computing environment to go effectively around this cycle.

1. XploRe - an interactive smoothing environment

The computing environment necessary to perform such experimental smoothing falls into three layers (Chambers, 1986):

α) the individual computer;

β) the operating system;

γ) the special logical structures for smoothing.

All three parts interact with each other. Since hardware α) and the system software β) that goes along with it has become affordable even for small institutions the discussion of what to choose for optimization of α) and β) does not seem too relevant to us. In fact we will present the system XploRe as it was developed on a "relatively simple" machine, an IBM AT. The data and program structures γ) for data smoothing and handling seem to be more important to achieve a high degree of interactiveness. They should fulfill the following basic requirements.

(1.1) The interactive system should allow convenient comparison of different fits, preferably in a graphical way.

(1.2) Certain viewpoints or snapshots (from different "angles") of the data and its smooth should be recordable.

(1.3) Results, summary statistics or verbalized impressions should be storable on the spot and visible at convenience.

(1.4) Intermediate stages of a smoothing analysis should be deletable or evocable. Input/Output to or via other layers of the computing environment must be possible.

(1.5) A dump and a reloading of the current stage of analysis should be possible.

In order to fulfill the above requirements we defined in XploRe the following basic objects:

vector,
workunit,
picture,
text.

Vectors are the simplest objects, they contain an alphanumeric data array of variable length. Workunits are collection of pointers to vectors and may include display and mask attributes. Picture objects are viewports, defining the location and tic marks of the axes in 2D or 3D views. Text objects are sequences of text lines with variable length.

In order to fulfill (1.4) and (1.5) we defined the following basic operations on these objects. Objects can be

created/deleted;
activated/deactivated;
read/written;
manipulated;
displayed.

The concept of the workunit object meets requirement (1.1). In its simplest form a workunit object can be thought of as a data matrix, but the actual realization as a record of pointers to existing vector objects makes it storage space economic. The additional feature of this object to include mask and display information makes exploratory techniques like brushing (Becker and Cleveland, 1986) easy to program. The display information as part of a workunit object makes it convenient to distinguish different functions: Whenever the workunit object is displayed (in a picture object) the corresponding display style information (part of this workunit) is used. This makes it easy to remember different curves. The mask part of this data object can be inherited to children objects (e.g. smooths) of a workunit and makes thus tracing of interesting points through several steps of an analysis possible, see Oldford and Peters (1986) for more information on this inheritance principle and this object oriented approach. A graphical description of workunits is depicted below in Figure 1.1.

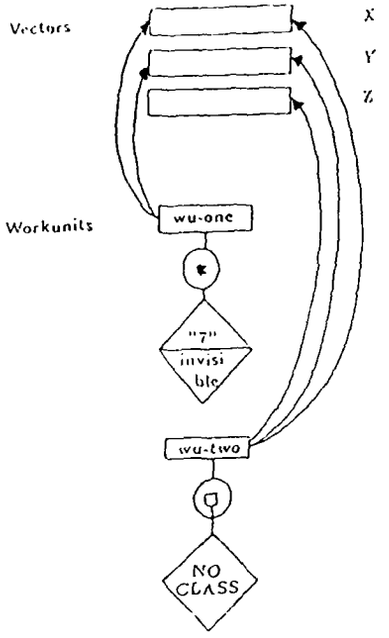


Figure 1.1. Two workunits with mask and display information

Figure 1.1 shows the situation where one wants to analyse a three dimensional data set consisting of vectors X, Y, Z. Workunit *wu-one* consists of the vectors X, Y, another *wu-two* points to all three vectors. When displaying *wu-one* one could have detected some interesting points, which one interactively has marked with the mask "7". Other observations might have been given the mask "invisible". Earlier one might have decided to see the remaining points as stars " * " (except those that have mask "7"). *Wu-two* is shown with square "□" and needles "|" pointing into the (X, Z) plane with no additional mask options.

Picture objects are designed to meet requirement (1.2) and certain information about the location of the 2D or 3D viewpart on the screen, the scaling of all the axes and the location of the axes on the physical screen. This object type is resident until its parts are changed. If one displays a workunit object and has found a reasonable scaling, this current picture object is evokable at later stages. A picture object can be graphically represented as in Figure 1.2.

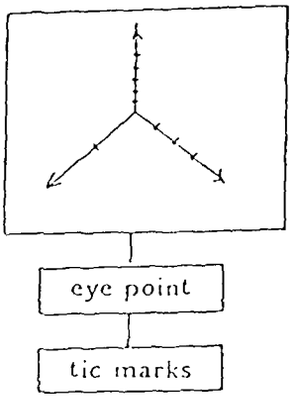


Figure 1.2. A picture object

Different workunits may be displayed in different picture objects. Figure 1.3 below shows a workunit (pointing to the raw data) as a pointcloud together with another workunit showing the smooth regression curve both in one picture object. A density estimate of the marginal density of X is displayed in another picture object (viewport "picture 2") at the upper right corner of the screen.

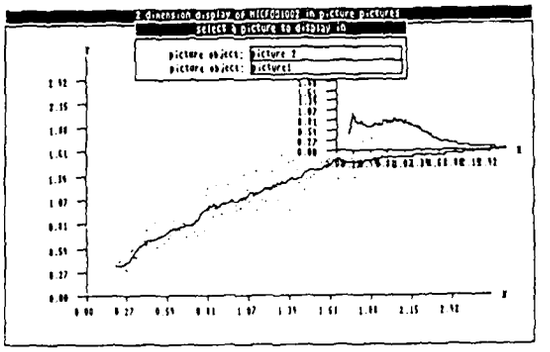


Figure 1.3. Two different picture objects

Text objects are defined according to (1.3). They contain ASCII text lines of variable column length. If such an object is displayed scrolling forward and backward in the actual text are possible. If a text object contains columns of data vectors (as ASCII information) it can be converted into a workunit object (with standard display and mask part) and vice versa.

2. Smoothing Techniques

The basic operations on the four objects have been defined above. All these operations are more or less self-explaining so that we concentrate in this section on the manipulation of workunit and picture objects. The different smoothing techniques entered via this manipulation of an active workunit are described below. The following lists are by no means exhaustive. XploRe (1987) is an open system, more soft work can be included, see section 3.

2.1 Regression Smoothing

- Regressogram (Tukey, 1961).
- k-nearest neighbor estimation (Mack, 1981).
- Supersmoothing (Friedman, 1984).
- Kernel smoothing (Nadaraya, 1964; Watson, 1964).
- WARPing (Härdle and Scott, 1988).
- Isotonic Regression (Barlow et al., 1972).
- Running Median (Tukey, 1977).
- Polynomial Regression (Shibata, 1981).
- Cross-validation (Clark, 1980).

2.2 Density Smoothing

- Histogram.
- k-nearest neighbor estimation (Cover and Hart, 1967).
- Kernel smoothing (Rosenblatt, 1956).
- (Log)Normal fitting.
- L_2 and Kullbach Leibler crossvalidation (Marron, 1987).

2.3 Additive Model

- Alternating Conditional Expectations (ACE) (Breiman and Friedman, 1985).
- Projection Pursuit Regression (PPR) (Friedman and Stuetzle, 1981).
- Recursive Partioning Regression Trees (RPR) (Breiman, Friedman, Olshen and Stone, 1984).
- Average Derivative Estimation (ADE) (Härdle and Stoker, 1988).

2.4 The interactive display

The interactive display features of XploRe allow manipulation of both workunit and picture objects. Removal, identification and classification of points is performed by pointing with a cursor to a group of points. This technique is incorporated in XploRe by the *label* and *mask* option of the graphics command menu, see Figure 2.1. The mask information will be inherited by the currently displayed workunit object. By clicking the "label" field the cursor can be moved to any point on the screen. After pressing ENTER a window pops up that shows the index of the observation (closest in Euklidean distance) together with the coordinate of the workunit. This feature enables the user to see all coordinates of a high dimensional workunit although he might be looking only at one "interesting" point in a two or three dimensional projection. The "mask" field allows the user to interactively define a rectangle of points which he would like to classify into groups 1-9 or invisible. The "unmask" option reverses this action. the *edit* field allows to change the ticmarks and the scaling of the axis and also the display style of the workunit currently shown. The *movoff* is a switch to *movon* which means that all screen information is stored in a movie fashion to disk. By pressing *movie* the saved screens will be shown, this feature allows tracking of past actions.

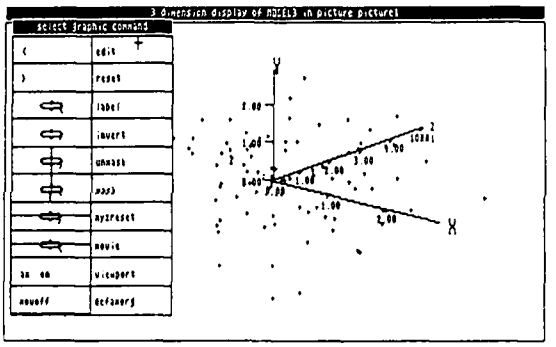


Figure 2.1. The interactive display

The *viewport* option allows the user to map certain sub-rectangles of the screen to the whole screen. The *defaxorg* field is for interactive definition of the axis origin. Clicking *ax on* switches to *ax off* which has the effect to display the data without the axis. The six fields above the axis control refer to rotations clock- and counterclockwise around each of the three axis in 3D space. The two fields in the upper left corner define the distance of the eyepoint relative to the pointcloud. Clicking successively ">" gives the impression to come closer to the data, whereas "<" makes the distance bigger. The 3D graphics have been programmed according to Newman and Sproull (1981).

The *edit* field is for locally changing the display style and for inheriting the current picture object ticmarks and axis labelling. Figure 2.2 shows the screen just after clicking "edit" in the situation of Figure 2.1.

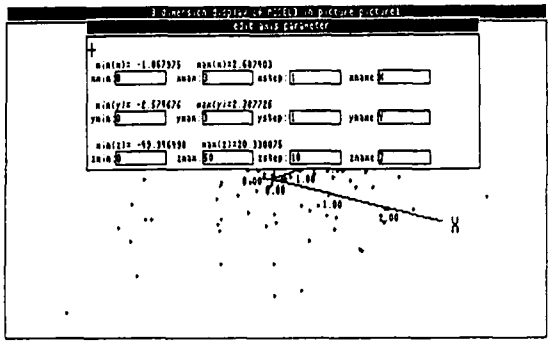


Figure 2.2. Editing the picture object.

The sensitive fields, shown by rectangles, show the current tics. By overwriting in these fields one changes the layout of the axis. The *reset* option gives a standard view in the cube $[0, max(x, y, z)]^3$.

2.5 Help information

Help files can be attached by the system programmer through a stack of "help windows". The designer of the computing environment determines at which analysis stage which "help windows" should appear. The help information is obtained by pressing F1. Subsequent pressing of the help key guides through the stack of currently attached help windows. The help windows are in fact internally handled as temporary text objects which are displayed as in Figure 2.3.

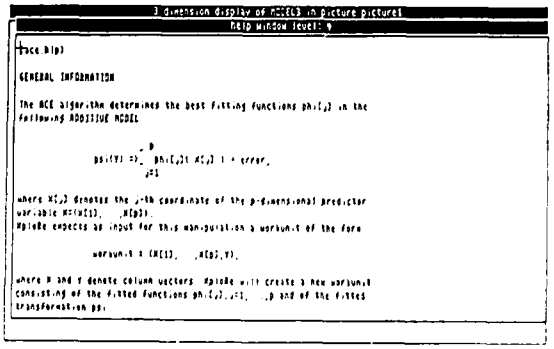


Figure 2.3. A help window

The help windows (and also text objects) can be scrolled backwards and forward by using the PgeDown and PgeUp key. All pulldown menus can be folded and unfolded by successive pressing of the F10 key.

3. Installing own procedures

The system XploRe can be enhanced by installing user written procedures. As an example of how to install own routines we describe how the *running median* primitive was implemented into XploRe. Assume that there is already a **procedure runmed** (y, n, k, s) with input array y, length n, smoothing parameter k and output array s (containing the running median sequence). An optimal algorithm has been given by Härdle , Reinholz and Steiger (1988). The user chooses this manipulation by mouseclicks and by definition the manipulation refers to the active workunit object. This workunit will then be temporarily sorted by the first column (interpreted as the predictor variable x), then the response variable y has to stripped off to determine the running median smooth s. It is convenient to build a vector object for this output array s and to create a workunit containing links (pointers) to the vector object containing the predictor variable x. In XploRe (respectively TURBO PASCAL) these operations would read as follows.

```
procedure dorunmed (wu);
var
   x,y,s: workarray;
   n,k: integer;
   xvec, yvec, svec, newwuobj: objectid;
begin
   quicksort(wu);
   getvector(wu, xvec, x, n, 1);
   getvector(wu, yvec, y, n, 2);
   getparameter(k); { reads the window size k
   from the keyboard }
   runmed(y, n, k, s);
   createobj(svec, vectorparttyp, "smooth");
    updatevector (svec, s, n);
    createobj(newwu, wuparttyp, "runmed");
    inclink(newwu, xvec);
    inclink(newwu, svec);
end;
```

The *getvector* procedure extracts from workunit *wu* the x and y array. The *createobj* procedure creates an object of the specified type (vectorparttyp, wuparttyp). The *updatevector (inclink)* procedure includes an array (a link) into vector objects (workunit objects).

Acknowledgement

I would like to thank Wolfgang Rossner who helped in the programming and design of XploRe. The system improved a lot through discussions with David Scott, Anders Holtsberg and Mark Aerts.

References

Barlow, R.E.; Bartholomew, D.J.; Bremner, J.M. and Brunk, H.D. (1972) Statistical Inference under Order Restrictions. *Wiley, London.*

Becker, R.A. and Cleveland, W.S. (1986) Brushing a Scatterplott Matrix: High-Interaction Graphical Methods for Analyzing Multidimensional Data. *Manuscript.*

Breiman, L.; Friedman, J.; Olshen, R. and Stone, C.J. (1984) Classification and regression trees. *Wadsworth, Belmont.*

Breiman, L. and Friedman, J. (1985) Estimating Optimal Transformations for Multiple Regression and Correlation. *J.Amer.Statist. Assoc., 80, 580–619.*

Chambers, J.M. (1986) Computing Environments for Quantitative Applications. *AT & T Bell Labs Stat. Research Reports No. 17.*

Clark, R. M. (1980) Calibration, Cross–validation and Carbon–14. II. *J.R.Statist. Soc. A 143, 177–194.*

Cover, T.M. and Hart, P.E. (1967) Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theory, 13, 21–27.*

Friedman, J. and Stuetzle, W. (1981) Projection Pursuit Regression. *J.Amer.Statist. Assoc., 76, 817–823.*

Friedman, J. and Tibshirani, R. (1984) The Monotone Smoothing of Scatterplots. *Technometrics, 26, 243–250.*

Härdle, W., Reinholz, A. and Steiger, W. (1988) Optimal Median Smoothing. *Manuscript.*

Härdle, W. and Stoker, T. (1988) Investigating smooth multiple regression by the method of average derivatives. *J.Amer.Stat.Assoc., submitted.*

Härdle, W. (1988) Applied Nonparametric Regression. *Book to appear.*

Härdle, W. and Scott, D.W. (1988) Weighted Averaging using Rounded Points. *Manuscript.*

Hastie, T. and Tibshirani, R. (1987) Generalized Additive Models: Some Applications. *J.Amer.Stat.Assoc., 82, 371–386.*

Ibragimov, I.A. and Hasminski, R.Z. (1982) Bounds for the Risk of Nonparametric Regression Estimates. *Theor. Prob.Appl., 27, 84–99.*

Mack, Y.P. (1981) Local Properties of k–NN Regression Estimates. *Siam J. Alg. Disc. Meth., 2, 311–323.*

Marron, J.S. (1986) Will the Art of Smoothing ever become a Science?. *in: Function estimates, (Marron, ed.) AMS Contemporary Mathematics 59.*

Marron, J.S. (1987) A Comparison of Cross-validation Techniques in Density Estimation. *Ann.Statist., 15, 152–162.*

McDonald, J. and Pederson, J. (1986) Computing Environments for Data Analysis: Part 3: Programming Environments. *Laboratory for Computational Statistics, Stanford Technical Report, 24.*

Newman, W.M. and Sproull, R.F. (1981) Principles of Interactive Computer Graphics. *Mc Graw-Hill.*

Oldford, R.W. and Peters, S.C. (1985) DINDE: Towards more Statistically Sophisticated Software. *MIT, Technical Report Tr-55.*

Rosenblatt, M. (1956) Remarks on some non–parametric estimates of a density function. *Ann. Math. Statist. 27, 642–669.*

Scott, D.W. (1986) Data Analysis in Three and Four Dimensions with Nonparametric Density Estimation. *in "Statistical Image Processing and Graphics" ed. E. Wegman, D. Priest, Marcel Dekker.*

Shibata, R. (1981) An Optimal Selection of Regression Variables. *Biometrika 68, 45–54..*

Silvermann, B.W. (1986) Density Estimation for Statistics and Data Analysis. *Chapman and Hall, London.*

Stone, C.J. (1977) Consistent Non–parametric Regression (with Discussion). *Ann. Statist. 5. 595–645.*

Stone, C.J. (1982) Optimal Global Rates of Convergence for Nonparametric Regression. *Ann. Statist., 10, 1040–1053.*

Tukey, J.W. (1961) Curves as Parameters and Touch Estimation.. *Proc 4th Berkeley Symposium, 681–694.*

Tukey, J.W. (1977) Exploratory Data Analysis. *Addison, Reading, Massachusetts.*

XploRe (1987) XploRe - a computing environment for eXploratory Regression and density smoothing. *Wirtschaftstheorie II, Universität D-5300 Bonn.*

INTERACTIVE MULTIVARIATE DENSITY ESTIMATION IN THE S LANGUAGE

David W. Scott and Mark R. Hall, Rice University

Abstract

We have been developing experimental software on workstations to produce high quality color graphical representations of multivariate density estimates via averaged shifted histograms. Some of our programs have been stand alone applications and some have been written in larger systems such as the S language. Part of our experiment was implementing our algorithms in Becker and Chambers' S interface language. We discuss our experiences and try to illustrate the results.

1. Introduction

We have been developing algorithms for data analysis with emphasis on graphical display, often innovative but non-standard in format. An example of this work has been the estimation and representation of nonparametric probability density estimates of data in R^d, $1 \le d \le 4$ (Scott and Thompson, 1983; Scott, 1985, 1986). Other examples include nonparametric regression, additive models, and computationally intensive algorithms such as cross-validation (Scott and Terrell, 1987; Scott, 1988; Härdle and Scott, 1988). All of the algorithms have been developed in Fortran (F77), with custom programming of an AED512 terminal for graphical output. However, in the past few years, we have increased our use of the S language (Becker and Chambers, 1984) for data manipulation and standard graphics.

The question we asked was: "Would it be both feasible and effective to use the S language for development of experimental algorithms on a UNIX workstation?" Some of these algorithms would have standard graphical output, such as x-y plots of cross-validation functions, while other algorithms would attempt to display three-dimensional density contours. Becker and Chambers (1985) have provided a mechanism by which any working F77 subroutine may be installed into the kernel, effectively becoming a "new" S function. This task is accomplished by writing an interface routine using a C-like language and calling S-supplied graphics calls inside the F77 routine. In fact, the built-in S functions themselves are written in the interface language with F77 subroutines for complex functions.

For anyone familiar with the S language (or other similar languages such as GAUSS on the PC), it is obviously desirable to have one's "established" routines available as S functions (Chambers, 1980). The primary benefits are on-line documentation, simplicity of input and output data handling, and device-independent graphics available both inside S functions or available for application on output S data structures. However, the process of creating and debugging interface routines can be more than a bit exciting even with established routines: does it make sense for *experimental* and *evolving* code?

After almost a year working with workstations, it seemed natural to evaluate that experience and plan the most productive strategy for our work. In addition to using S, two other approaches were considered. First, the "old way" of writing large custom Fortran routines on a mainframe or workstation with output to an AED 512 terminal with byte-level control or with output to an IRIS terminal controlled by calls to high-level graphics libraries furnished by Silicon Graphics. The second approach is similar to the first but uses an integrated platform such as a Sun 3/60 workstation or a Mac II and a language such as Fortran, Pascal, or C. The second approach was the only alternative seriously considered.

There are several advantages of writing high level language routines directly rather than using S. The codes tend to be smaller and a bit faster. S functions take much longer to compile during the debugging phase. Moreover, there is less code to debug. In particular, S programming creates intermediate Fortran files that generate errors that must be traced back to the original interface and Fortran routines. This traceback problem is familiar to users of the Unix Fortran preprocessor, Ratfor.

The advantages of using S are several. It is much easier to prepare test data and input using the wide array of available S functions and data structures. The calling sequences for S functions are much shorter than the actual Fortran subroutines, since only the input variables need be specified (output variables are automatically returned in a data structure) and many input variables can be given common default values. It is also very easy to create quick and dirty graphs for experimentation and then modify and improve the graphs. But the most important reason is that coding experimental routines in S minimizes *software loss*. Have you ever tried to run an "experimental" code after a six month layoff and get anything useful from it? In the summer of 1987, I developed a code in S for computing average derivative estimates (Härdle and Scott, 1988) while visiting Wolfgang Härdle in Bonn. A year later, while Härdle was an invited lecturer at Rice, we were able to immediately use those routines on new data he brought "cold." I have seldom had that experience in any other language. So a significant part of the advantage is that output can be well-organized by design to reside in S data structures that can be interactively listed, graphed, or analyzed. Too often a directory with an experimental Fortran algorithm contains a bunch of files named fort.1, fort.2, etc. Another S tool that helps minimize lost work is the *diary* file, which contains a record of all S sessions. Thus the advantages of using S relative to custom routines are significant for an experienced developer.

Of course, S is not totally unique among languages with respect to these capabilities. In fact, John McDonald and others prefer a totally unified environment such as that offered on a Symbolics workstation (McDonald and Pedersen, 1985). It is clear the such a LISP platform is powerful but S provides more immediate productivity gains since it has a feel more similar to classical languages.

2. Data Analysis via Density Estimation

The symmetric positive kernel estimate studied by Rosenblatt (1956) and Parzen (1962) has been widely used to study data $x_1, x_2, \ldots, x_n$:

$$f_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

This formula is easily extended to multivariate data by using a multivariate probability density function as the kernel. A much more convenient and computationally inexpensive form is the averaged shifted histogram (Scott, 1985):

$$\hat{f}(\cdot) = \frac{1}{m} \sum_{k=1}^{m} H(h; \frac{k}{m}h)(\cdot),$$

where $H(h;t)(\cdot)$ is an equally spaced histogram with bin width h and mesh location uniquely determined by having one mesh node at t. The ASH amounts to a weighted average of rounded points (WARP) and is also easily extended into several dimensions. This idea can also be applied to a wide array of nonparametric and additive models (Härdle and Scott, 1988).

2.1 Representation of Density Estimates

The most effective way to represent multivariate density estimates has been a source of many interesting discussions and much research. In the case of bivariate data, we have heard talks in which perspective views of a three dimensional bivariate density surface have been severly criticized relative to contour plots. Such a position seems far too extreme. However, for our purposes, contour

plots are preferable because they extend naturally into higher dimensions. The display methodology we have advocated (Scott and Thompson, 1983; Scott, 1983; and Scott, 1986) has been to draw α-level contours where $0 \leq \alpha \leq 1$. Specifically, these contours, which we shall refer to as "α-shells", are defined by the sets

$$\mathbf{S}_\alpha = \{(\mathbf{x} \in \mathbf{R}^d : \hat{f}(\mathbf{x}) = \alpha \hat{f}(\mathbf{m})\}$$

where $\mathbf{m}$ is the mode of $\hat{f}$:

$$\mathbf{m} = \arg \max_{\mathbf{x} \in \mathbf{R}^d} \hat{f}(\mathbf{x}).$$

For trivariate data, there is one degree of freedom, namely the density level α, although the viewing angle might reasonably be considered another degree of freedom. For quadravariate data $\mathbf{x} = (x, y, z, t)$, we display the trivariate shell satisfying

$$\mathbf{S}_\alpha(t_0) = \{(x, y, z) \in \mathbf{R}^3 : f(x, y, z, t | t = t_0) = \alpha \hat{f}(\mathbf{m})\}.$$

Clearly there are two degrees of freedom, the contour level α and the slice intercept t_0. For those interested in animation of algorithms, both parameters provide for interesting data views. We have in fact created movies of four-dimensional LANDSAT data using this technique (Scott and Jee, 1984).

2.2 Example with Particle Physics Data Set

We begin by examining a four-dimensional particle physics data set provided with the S language (Friedman and Tukey, 1974). These data fall in a relatively narrow strip on the fourth variable, as can be seen by examining all pairwise scatter diagrams of these data in Figure 1. This plot was constructed by the *pairs* S command. This conclusion is strengthened by an examination of the four marginal histograms in Figure 2. Since there are 500 points in each of the scatter diagrams, there is a great deal of overlap in the plot. We can examine the "V" structure in the variable 2-3 plot by computing a bivariate averaged shifted histogram, which is shown in Figure 3 (Scott, 1987). The density plot indicates this is not a symmetric "V" and fewer points fall along the two rays than the scatter diagram may suggest. If we slice the variable 4 axis into ten bins and then focus on the last bin where $t = t_{10}$ (which contains most of the data), we can examine the density shell $\mathbf{S}_{20\%}(t_{10})$, that is, the 20% contour shell of a slice of the quadravariate density estimate in the last bin along the variable 4 axis. The other axes were binned into thirty bins. The continuous shell surface in three dimensions is represented by a collection of two dimensional contour slices perpendicular to the x and y axes. In Figure 5, we have superimposed a higher level density contour. From these figures, we conclude the data are rather uniformly distributed along this truncated "U" shaped core.

2.3 Example with Transformed Particle Physics Data Set

Kernel methods are sensitive to data that fall or nearly fall into subspaces. Thus it may be appropriate to transform the original data in such a manner that the resulting marginal distributions are much less skewed. We have done this with the particle physics data using the respective marginal transformations: $\log(x_1)$, $\sqrt{\log(1 + x_2)}$, $-\sqrt{\log(1 - x_3)}$, and $-\sqrt{\log(1 - x_4)}$. The effect of this transformation is shown in Figures 6, 7, and 8. The structure in these data is fairly clear. It is interesting to consider the efficacy of the transformation in this case. Without specific reference to the original purpose of these data, it is difficult to say more.

3. Marching Cubes Display of Density Shells

Oftimes it is desirable to have a high resolution view of a particular density shell surface. The previous representation is more useful for providing a rather broad brush view of the entire density by examining several density shells simultaneously. However using techniques used in CAD-CAM applications, the ASH shells may be rendered as shaded solids. A particularly nice formulation of this idea may be found in Lorensen and Cline (1987). They call their method "marching cubes." A three dimensional triangularization is computed and then displayed using a false-color algorithm based upon the direction of a unit normal. In the authors' application, a further color smoothing was desired and was accomplished by a Gouraud shading technique. Such a technique on the averaged shifted histogram might also be desirable, but we have chosen not to do so to emphasize the piecewise linear nature of the estimator.

In Figure 9, we show a screendump of the triangularization of an exact trivariate normal density with covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & .8 & .8 \\ .8 & 1 & .8 \\ .8 & .8 & 1 \end{pmatrix}.$$

This is a 30 by 30 by 30 mesh and the display is at the 5%-level. Notice the visual discontinuity is really rather small even with such a relatively coarse binning. On advanced color hardware, these surfaces can be rotated in near real time. However, the number of triangles is so large that a 10-20 MIPS workstation is necessary for real time rotation.

The data discussed in sections 2.2 and 2.3 were also examined using this tool. The ASH estimates computed in the S function were written to an ASCII file and then input to a C suntools program on a color Sun 3/260 workstation. While this was somewhat cumbersome, it was very efficient from an experimental point of view. In Figure 10, we show the triangularization of the 10%-level of the ASH of the untransformed particle physics data. In Figure 11, we show the 5%-level of the ASH of the transformed particle physics data. Such plots provide a great amount of detail at one contour level, but not at several levels as before. However, several advanced graphics workstations provide for transparent views of surfaces. Displaying and rotating several ASH contours levels simultaneously is the authors' dream.

4. Acknowledgments

The research of the first author was supported in part by the Office of Naval Research and the Army Research Office under contracts N00014-85-K-0100 and DAAG-29-82-K0014, respectively.

5. References

Becker, R.A. and Chambers, J.M. (1984), *S: An Interactive Environment for Data Analysis and Graphics* Wadsworth, Belmont, California.

Becker, R.A. and Chambers, J.M. (1985), *Extending the S System* Wadsworth, Belmont, California.

Chambers, J.M. (1980), "Statistical Computing: History and Trends," *The American Statistician* 34:238-243.

Friedman, J.H. and Tukey, J.W. (1974), "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Trans. Comput.* C-23:881-889.

Lorensen, W.E. and Cline, H.E. (1987), "Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm," *ACM Computer Graphics* 21:163-169.

Härdle, W. and Scott, D.W. (1988), "Smoothing in Low and High Dimensions by Weighted Averaging Using Rounded Points," Rice University Technical Report 88-16.

McDonald, J.A. and Pedersen, J. (1985), "Computing Environments for Data Analysis II: Hardware," *SIAM J. Scientific and Statistical Computing* 6:1013-1021.

Parzen, E. (1962), "On Estimation of a Probability Density Function and Mode," *Ann. Math. Statist.* 33:1065-1076.

Rosenblatt, M. (1956), "Remarks on Some Nonparametric Estimates of a Density Function," *Ann. Math. Statist.* 27:832-837.

Scott, D.W. (1983), "Nonparametric Probability Density Estimation for Data Analysis in Several Dimensions," *Proceedings of the Twenty-Eighth Conference on the Design of Experiments in Army Research Development and Testing*, pp. 387-397.

Scott, D.W. (1985), "Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions," *Annals of Statistics* 13:1024-1040.

Scott, D.W. (1986), "Data Analysis in 3 and 4 Dimensions With Nonparametric Density Estimation," in *Statistical Image Processing*, E.J. Wegman and D. DePriest, Eds., Marcel Dekker, New York, pp. 291-305.

Scott, D.W. (1987), "Software for Univariate and Bivariate Averaged Shifted Histograms," Rice Technical Report 311-87-1.

Scott, D.W. (1988), "Comment on paper by Härdle, Hall, and Marron," *J. American Statistical Association*, 83, 96-98.

Scott, D.W. and Jee, R. (1984) "Nonparametric Analysis of Minnesota Spruce and Aspen Tree Data and Landsat Data," *Proceedings of the Second Symposium on Mathematical Pattern Recognition and Image Analysis*, pp. 27-49.

Scott, D.W. and Terrell, G.R (1987), "Biased and Unbiased Cross-Validation in Density Estimation," *J. American Statistical Association* 82:1131-1146.

Scott, D.W. and Thompson, J.R.(1983), "Probability Density Estimation in Higher Dimensions," *Proceedings of the 15th Symposium on the Interface of Computer Science and Statistics*, J.E. Gentle, Ed., North-Holland, Amsterdam, pp. 173-179.

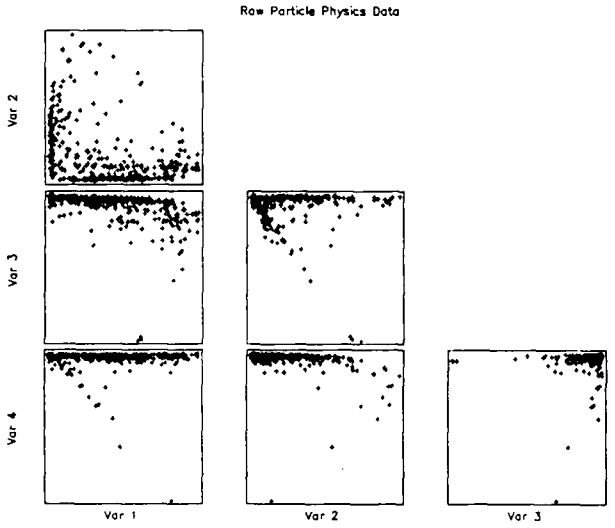


Figure 1. Pairwise scatterplots of the particle physics data.

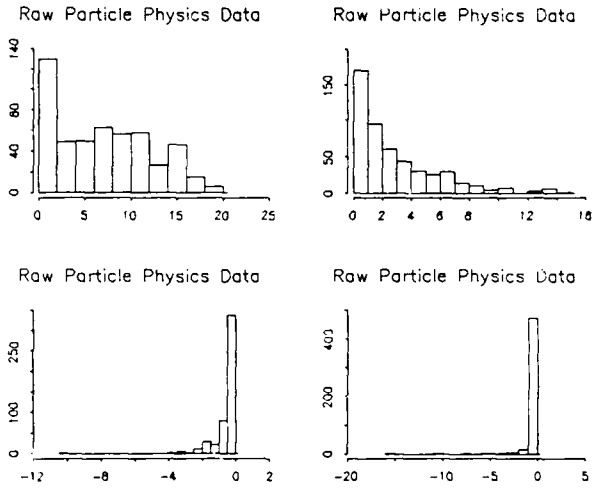


Figure 2. Histograms of particle physics data variables. Variables 1 and 2 are in the first row and 3 and 4 in the second.

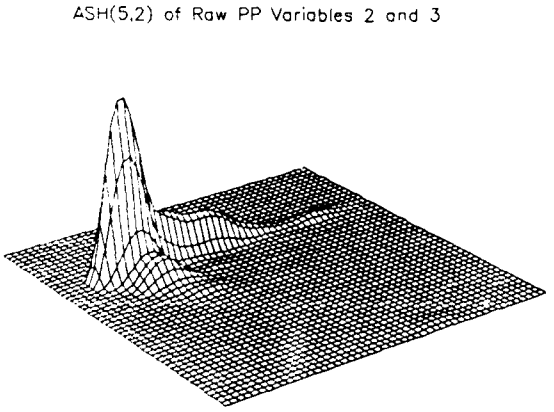


Figure 3. Perspective plot of a bivariate averaged shifted histogram estimate of variables 2 and 3. The smoothing parameters are given in parentheses.

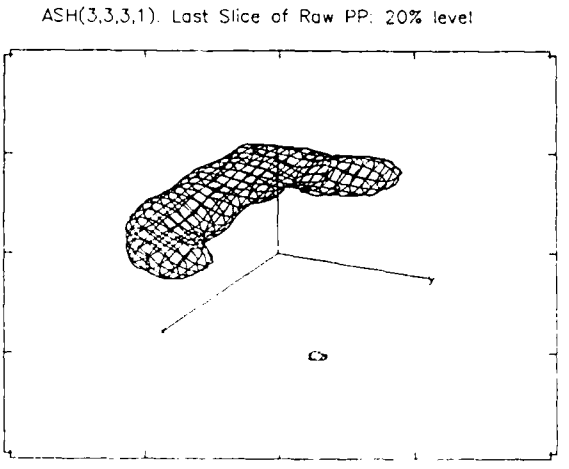


Figure 4. The $\alpha = 20\%$ shell slice of the estimated quadravariate averaged shifted histogram.

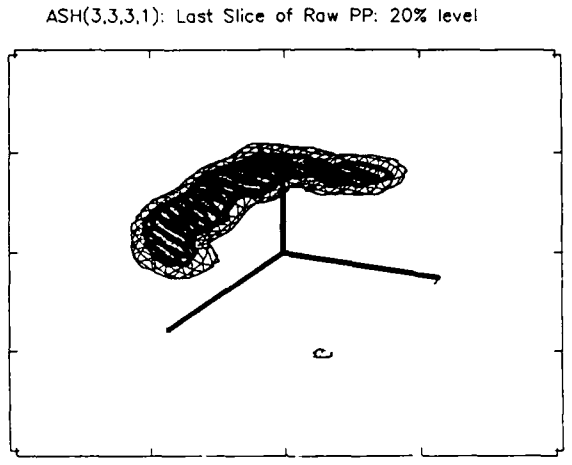


Figure 5. In this figure, we have superimposed the $\alpha = 50\%$ density shell onto Figure 4.

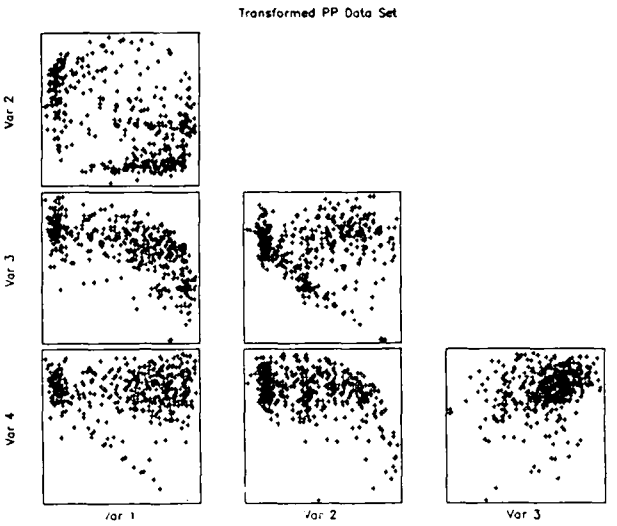


Figure 6. Scatterplot matrix of transformed particle physics data.

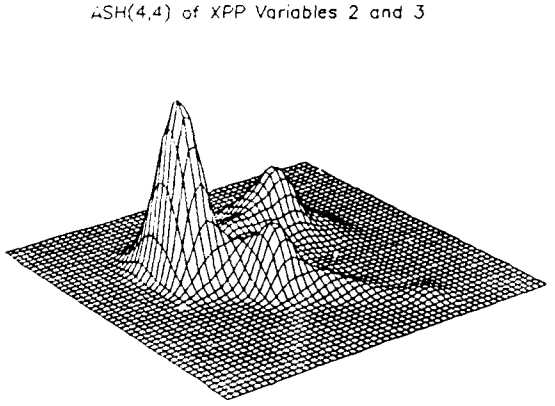


Figure 7. Perspective plot of a bivariate averaged shifted histogram estimate of the transformed variables 2 and 3. The smoothing parameters are given in parentheses.

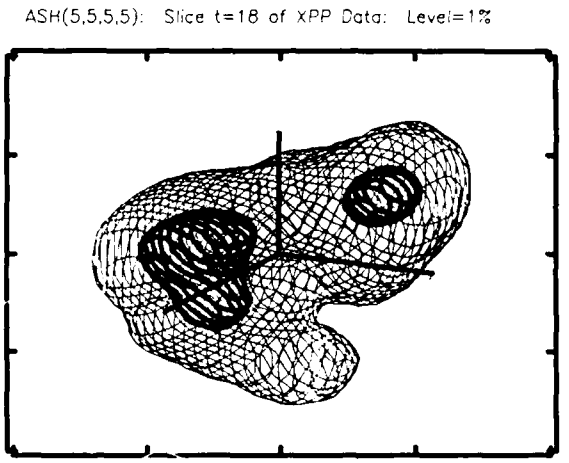


Figure 8. The $\alpha = 1\%$ and $\alpha = 40\%$ shell contour slices of the estimated quadravariate ASH of the transformed particle physics data, sliced at about the middle of the fourth variable.

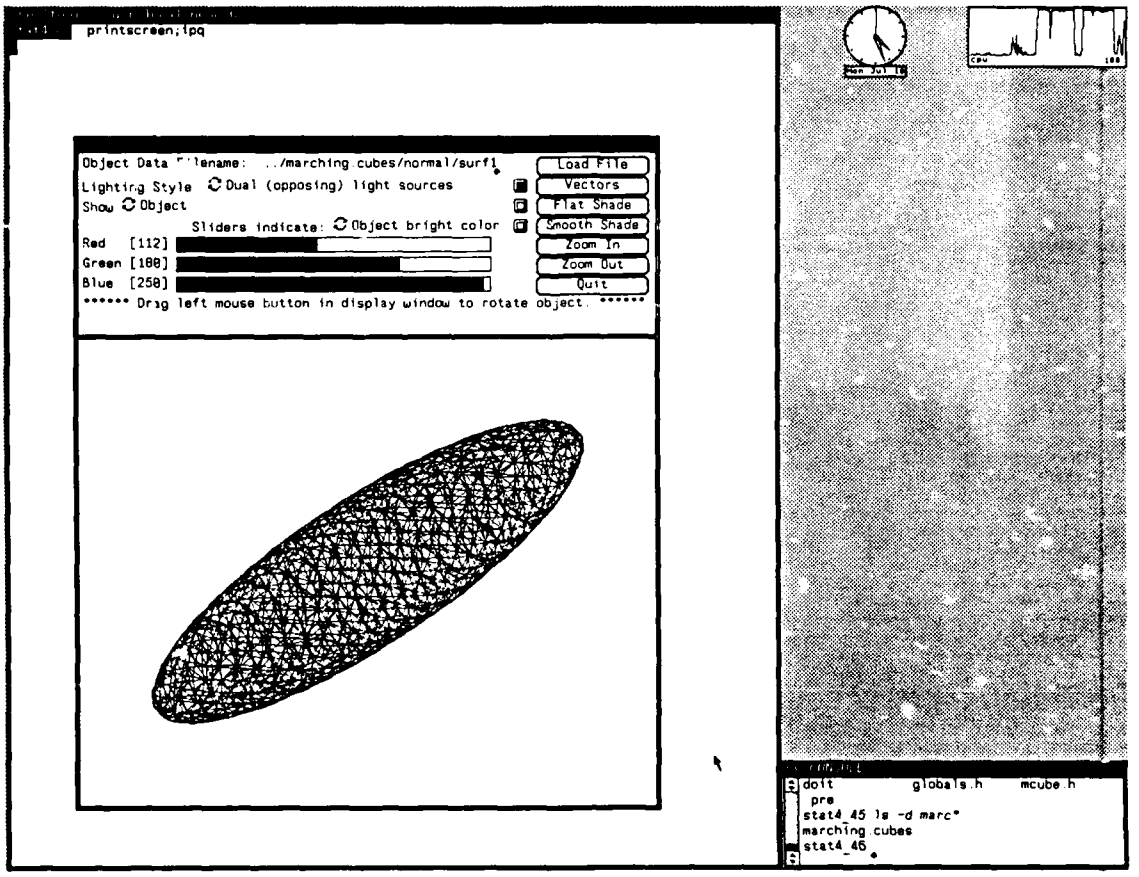


Figure 9. Triangularization of an $\alpha = 5\%$-shell of a trivariate normal density with correlations = 0.8.

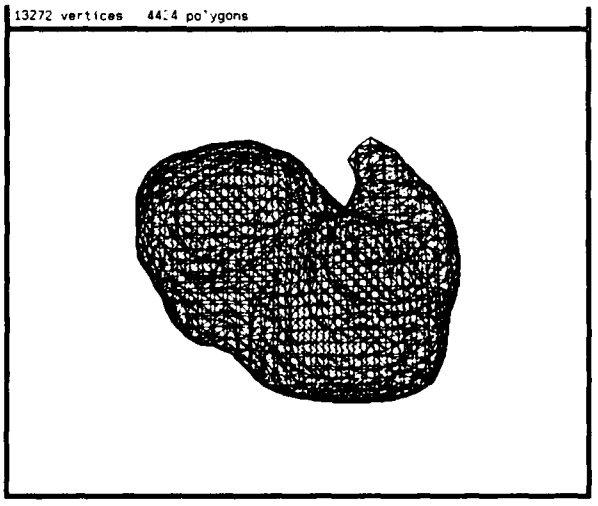


Figure 10. Triangularization of an $\alpha = 10\%$-shell slice of the raw particle physics data.

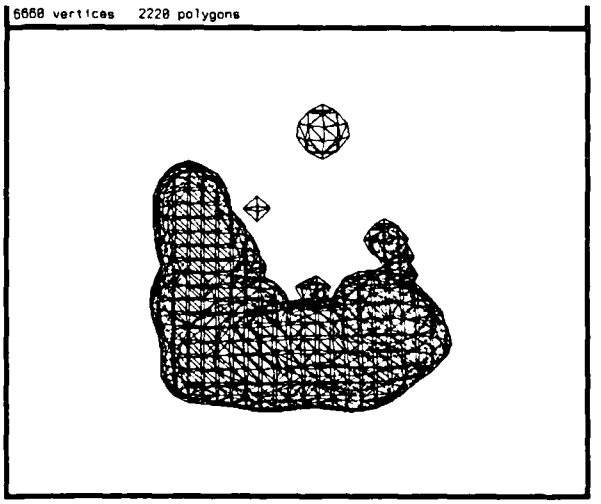


Figure 11. Triangularization of an $\alpha = 5\%$-shell slice of the transformed particle physics data.

SMOOTHING DATA WITH CORRELATED ERRORS

N. S. Altman

Biometrics Unit

Cornell University

ABSTRACT

Kernel smoothing is a common method of estimating the mean function in the nonparametric regression model

$$y = f(x) + \epsilon$$

where $f(x)$ is a smooth deterministic mean function, and ϵ is an error process with mean zero. In this paper, the mean square error of kernel estimators is computed for processes with correlated errors, and the estimators are shown to be inconsistent under restrictive assumptions on the sequence of error processes. The standard techniques for bandwidth selection, such as cross-validation and generalized cross-validation, are shown to perform very badly when the errors are correlated. Standard selection techniques are shown to favor undersmoothing when the correlations are predominantly positive, and oversmoothing when negative. However, the selection criteria can be adjusted to correct for the effect of correlation.

Method of moments estimates of the correlation function based on residuals are shown to be consistent when the bandwidth is chosen in such a way that the smooth is consistent. However, in finite samples, oversmoothing leads to estimates of correlation which are too large, while undersmoothing leads to estimates which are too small.

Keywords mean squared error, kernel smoothing, correlated errors, bandwidth, cross-validation, generalized cross-validation

1. Introduction

Kernel smoothing is a common method of estimating the mean function in the nonparametric regression model

$$y = f(x) + \epsilon \tag{1}$$

where $f(x)$ is a smooth deterministic mean function, and ϵ is an error process with mean zero. The focus of this work is on estimating the unknown mean function when the design points are equally spaced on $[0,1]$, and the errors come from a stationary correlated process. The kernel estimators of Priestley and Chao (1972) are used. These have the form

$$\hat{f}_{\lambda,n}(x) = \sum_{j=0}^{n} w_\lambda(x,j) y_{n_j}$$

where the weights are

$$w_\lambda(x,j) = \frac{K(\frac{x - x_{n,j}}{\lambda})}{n\lambda}.$$

K is called the kernel function, λ is a smoothing parameter, called the bandwidth, and n is the sample size.

Only kernels with the following properties are considered:

A) K is symmetric about 0.

B) K has support only on the interval $(-\frac{1}{2}, \frac{1}{2})$.

C) K is Lipschitz continuous of order $\alpha > 0$.

K is called a kernel of order p if all the first $p-1$ moments of K are 0, and the p^{th} moment,

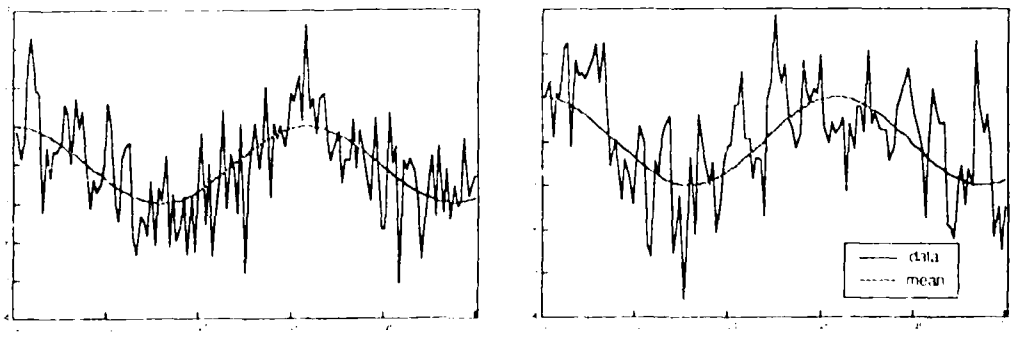
$$\nu_K = \int x^p K(x)dx$$

is not zero. The squared norm of K

$$W_K = \int K^2(x)dx \tag{2}$$

is also needed.

For sample size n, the observations $y_{n,0}, \ldots, y_{n,n}$ are assumed to be generated by the nonparametric regression



(a) Normal(0,1) errors (b) AR(1) errors, $\rho = .5$

Figure 1: *Raw data and mean function for $y = \cos(3.15\pi x) + \epsilon$, where the error variance is 1.0*

model (1), with observations taken at design points $x_{n,i} = \frac{i-1}{n-1}$. The errors are assumed to come from a stationary process with covariance function

$$E(\epsilon_{n,i}, \epsilon_{n,j}) = \sigma^2 \rho_n(|i-j|).$$

where the variance, σ^2, is independent of n and $\rho_n(k)$ is a correlation function depending on n. The variance matrix of the errors will be denoted by Σ_n.

The purpose of this paper is to explore the properties of kernel smoothers, and the use of model selection techniques, such as cross-validation (CV), (Allen,1974, Stone, 1974 and Geisser, 1975), and generalized cross-validation (GCV) (Craven and Wahba, 1979) when the errors are not independent, but instead come from a stationary correlated process.

Figures 1a and 1b show realizations of the process $y = cos(3.15\pi x) + \epsilon$ when the errors come from, respectively, a Gaussian process with unit variance, and an AR(1) process with the same variance and $\rho = .5$. The Gaussian process used in Figure 1a was used to generate the shocks for the AR(1) process in Figure 1b, so the resulting sample paths are very similar. Figures 2a and 2b show kernel estimates of the mean function for this data when the bandwidth was chosen using CV. For the realization with independent errors, the estimate is quite smooth and captures the main features of the mean function. For the realization with correlated errors, the estimate is far too wiggly. Figures 3a and 3b show the estimates of the mean function for this data using the optimal (minimum totalled squared error) value of the bandwidth. The estimates for the independent and AR(1) realizations are now quite similar, and capture the main features of the true mean function.

This paper addresses some of the issues raised by this example. How good are kernel and nearest neighbor smoothers for nonparametric regression estimation when the errors are correlated? How is the performance of standard model selection techniques effected by correlation of the errors? Can better model selection techniques be devised for use with correlated data?

2. Mean Square Error

We [illegible] assumed to have finite second [illegible]

ments, the MSE, defined by

$$MSE(x, \lambda, n) = E(\hat{f}_{\lambda,n}(x) - f(x))^2$$

is often used as a goodness of fit criterion and as a means of assessing the asymptotic properties of the estimators. The optimal smoothing parameter is often considered to be the one which minimizes the MSE totalled, or equivalently, averaged, over the design points (TSE and ASE, respectively).

The $MSE(x, \lambda, n)$ is

$$MSE(x, \lambda, n) = B^2(x, \lambda, n) + V_1(x, \lambda, n).$$

where the bias term is

$$B(x, \lambda, n) = w'_\lambda(x, \bullet)(f(\bullet) - f(x)) \qquad (3)$$

and the variance term is

$$V_1(x, \lambda, n) = w'_\lambda(x, \bullet)\Sigma_n\, w_\lambda(x, \bullet).$$

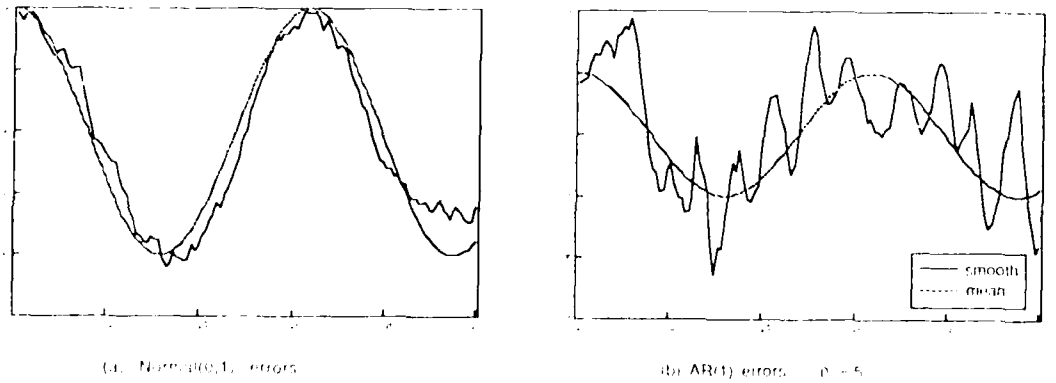
Notice that the bias depends on the sample size only via the selection of design points and is not affected by the correlation structure. For mean functions with at least p derivatives, and kernels of order p, Gasser and Müller (1979) computed the asymptotic form of the squared bias (when the design points become dense on the interval) to be

$$B^2(x, \lambda, n) = (\lambda^p s_K f^{(p)}(x)/p!)^2 + o(\lambda^{2p}) + o(\frac{1}{n})$$

when $\lambda \to 0$ and $n\lambda \to \infty$ and $\frac{\lambda}{2} < x < 1 - \frac{\lambda}{2}$. Kernel estimators are asymptotically unbiased under these conditions.

Correlation of the errors affects the the variance term very strongly. When the errors are independent, the variance term is $\sigma^2 \| w_\lambda(x, \bullet) \|^2$, where $\|\bullet\|$ is Euclidean norm. As the design points become dense on the interval, $\| w_\lambda(x, \bullet) \|^2 \to \frac{w_K}{n\lambda}$, so the behavior of the variance function decreases as $O(1/n\lambda)$ regardless of the shape of the kernel. When the errors are correlated, the behavior of the variance term as a function of the bandwidth depends on both the correlation function and the kernel.

Figure 4 is a plot of $V_1(x, \lambda, n)/\sigma^2$ for the uniform kernel with autoregressive errors of order 1 (AR(1)) with various values of $\rho(1)$. The variance term still decreases with n, and has a similar shape to the independent case, but the rate of decrease is much slower when $\rho(1) > 0$ and is much faster



(a) Normal(0,1) errors (b) AR(1) errors, $\rho = .5$

Figure 2: *Smoothed estimate and mean function for $y = cos(3.15\pi x) + \epsilon$. Cross-validation was used to pick the smoothing parameter.*

when $\rho(1) < 0$. Since the bandwidth controls MSE by trading off variance for bias, this suggests that, compared to the independent case, larger bandwidths will be needed when the correlations are positive, and smaller bandwidths when the correlations are negative. This is also suggested by Figure 5, which shows a typical realization of a process from model (1) when the errors are correlated. When the correlation is positive, nearby errors tend to have the same sign, and a large bandwidth is needed to average them out. When the correlation is negative, the errors fluctuate rapidly in sign, and only a small bandwidth is needed. Since larger bandwidths lead to larger bias, this also implies that, at the optimal bandwidth, the MSE will be larger correlations are positive, and smaller when the correlations are negative.

Explicit evaluation of $V_1(x_{n,i}, \lambda, n)$ makes these ideas precise. The critical statistic is the sum of the correlations (when it exists)

$$S_{\rho_n} = \sum_{j=1}^{\infty} \rho_n(j).$$

Theorem 1: *If the correlations satisfy*

$$\frac{1}{n\lambda} \sum_{j=1}^{[n\lambda]} j|\rho_n(j)| = o(1)$$

and the kernel function satisfies conditions A - C, then the correlations are absolutely summable and

$$V_1(x_{n,i}, \lambda, n) = \sigma^2 \frac{W_K}{n\lambda}(1 + 2S_{\rho_n}) + o(\frac{1}{n\lambda})$$

where W_K was defined by (2).

Proof of this theorem involves replacing Σ_n by the circulant matrix

$$R(i,j) \quad \begin{cases} \sigma^2 \rho(|i-j|) & |i-j| \le [\frac{n}{2}] \\ \sigma^2 \rho(n - |i-j|) & |i-j| > [\frac{n}{2}]. \end{cases}$$

The condition on the correlations makes $n(x_{n,i}, \bullet)'(\Sigma_n - R)(x_{n,i}, \bullet)$ negligible, and the result follows from the Lipschitz continuity and symmetry of the kernel. The theorem is also true for suitably chosen boundary kernels. The details are in Altman 1987 and 1988.

One consequence of this theorem is that kernel estimators can be consistent only if S_{ρ_n} is bounded as $n \to \infty$. This condition is clearly not satisfied if the errors have been generated by a weakly continuous stochastic process, $\rho_n(\varepsilon_i, \varepsilon_j) = \rho(\frac{|i-j|}{n})$. This process has been discussed by Hart and Wehrley (1986) and Parzen (1959 and 1961). An important result from these papers is that, if only a single realization of the process has been observed, there are no consistent linear estimators of the mean function as the design points are sampled more and more densely on the unit interval. Parzen's results show that the only unbiased linear estimator of $f(x)$ is y_x (with variance σ^2). Hart and Wehrley compute the bias and variance of kernel estimators, and show that, despite the lack of consistency, considerable improvements (in terms of mean squared error) can be made by using kernel estimators with $\lambda > 0$.

A second consequence of this theorem is that, if $S_{\rho_n} \to S_\rho$ as $n \to \infty$, kernel estimators behave as they would with independent errors with a different variance term. This is expressed in Corollary 1.1, below.

Corollary 1.1: *Suppose $S_{\rho_n} \to S_\rho$. Let z_x be the process generated by*

$$z_x = f(x) + u_x$$

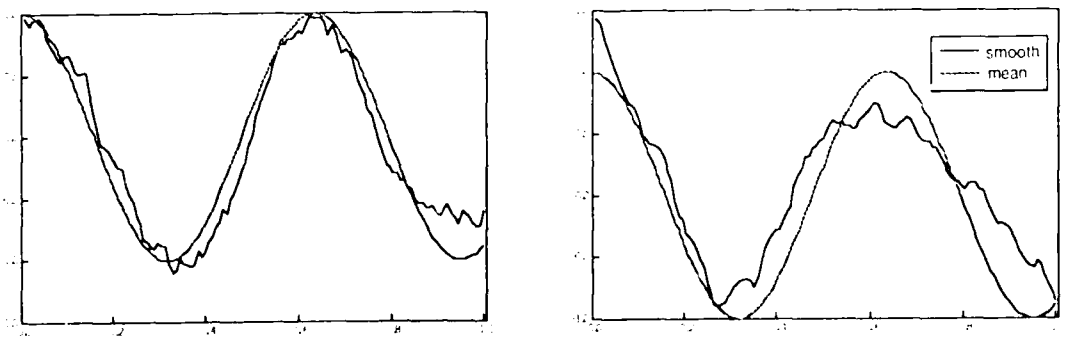
where the errors u_x are independent with variance $\sigma^2(1 + 2S_\rho)$, and z has the same mean function as y. Then, under the conditions of Theorem 1, asymptotically, as $\lambda \to 0$ and $n\lambda \to \infty$,

$$\frac{MSE_y(x_{n,i}, \lambda, n)}{MSE_z(x_{n,i}, \lambda, n)} \to 1.$$

As a consequence, the asymptotically optimal bandwidth, $\lambda(y)$, for estimating f from y is the same as estimating from z. Let Z_x be the process generated by

$$Z_i = f(x) + U_i$$

where the errors U_i are independent with variance σ^2 and let $\lambda(Z)$ be the asymptotically optimal bandwidth for estimating f from Z. If $S_{\rho_n} > 0$, then $\lambda(y) > \lambda(Z)$ and $MSE_y(x_{n,i}, \lambda(y), n) < MSE_y(x_{n,i}, \lambda(Z), n)$. If $S_{\rho_n} < 0$, then $\lambda(y) < \lambda(Z)$ and $MSE_y(x_{n,i}, \lambda(y), n) < MSE_y(x_{n,i}, \lambda(Z), n)$. Also kernel estimation is consistent under these conditions.



(a) Normal(0,1) errors (b) AR(1) errors $\rho = .5$

Figure 3: *Smoothed estimate and mean function for y=cos(3.15πx)+ε. Minimum totalled squared error was used to pick the smoothing parameter.*

Corollary 1 also shows that the results of Gasser and Müller (1979) about the shapes of optimal kernels continue to hold when the errors are correlated.

3. Selecting a Smoothing Parameter

For a given, finite set of observations, choice of an effective smoothing parameter is of considerable interest. A "good" value of the smoothing parameter will result in a small value of $MSE(x_{n,i}, \lambda, n)$.

Several criteria based on the data have been used for bandwidth selection. Those most commonly used are CV, (Allen,1974, Stone, 1974 and Geisser, 1975), GCV (Craven and Wahba, 1979), and Mallows C_L (Mallows, 1973). The properties of these criteria, including convergence of the smoothing parameter chosen by one of the selection criteria to the truly optimal value, and the asymptotic equivalence of the the criteria, have been explored in some detail for the independent case. As Figure 2 demonstrates, these criteria perform well when the errors are independent, but perform very poorly when the errors are correlated.

CV is based on the "deleted" residuals, $y_{n,i} - f_{\lambda,n,i}(x)$ where $f_{\lambda,n,i}(x)$ is the estimator which does not use $y_{n,i}$. Figure 6 provides a heuristic argument for the failure of CV when the errors are positively correlated. In this case, the errors for data near $x_{n,i}$ are tend to have the same sign as the error of $y_{n,i}$. As a result, $f_{\lambda,n,i}(x)$ lies closer to $y_{n,i}$ than $f(x)$ and the "deleted" residual is too small as an estimator of the true error. As a result, CV underestimates the variance of the estimator, and tends to pick bandwidths which are too small. As Theorem 2 demonstrates, the converse is also true. If the correlations are negative, CV overestimates the variance, and tends to pick bandwidths which are too large.

Mallows' C_L, CV and GCV can all be viewed as estimators of squared prediction error, based on a correction to the observed squared residual. The prediction error at a point, $x_{n,i}$, is the difference between a putative new realization of the process, and the smooth based on the actual observations. The errors in the new observations are independent of errors in the original observations, so the expected squared prediction error (ESPE) is

$$\begin{aligned} ESPE(x_{n,i}, \lambda, n) &= E\left(y_{new}(x_{n,i}) - \hat{f}_{\lambda,n}(x_{n,i})\right)^2 \\ &= \sigma^2 + MSE(x_{n,i}, \lambda, n). \end{aligned}$$

The residual at $x_{n,i}$,

$$r(x_{n,i}, \lambda, n) = y_{x_{n,i}} - \hat{f}_{\lambda,n}(x_{n,i})$$

is a natural estimator of prediction error, but the squared residual is biased as an estimator of $ESPE(x_{n,i}, \lambda, n)$ as it has expectation

$$\begin{aligned} E(r^2(x_{n,i}, \lambda, n)) = \sigma^2 + MSE(x_{n,i}, \lambda, n) \\ - 2\sigma^2 w_\lambda(x_{n,i}, i) - V_2(x_{n,i}, \lambda, n). \end{aligned} \tag{4}$$

The term $2\sigma^2 w_\lambda(x_{n,i}, i)$ arises because $y_{x_{n,i}}$ is both a term in the estimator, $\hat{f}(x_{n,i})$ and the estimator of $y_{new}(x_{n,i})$. The additional variance term,

$$V_2(x_{n,i}, \lambda, n) = 2\sigma^2 \sum_{j \neq 0} w_\lambda(x_{n,i}, i + j)\rho_n(j).$$

arises because of the correlation between $e_{n,i}$ and the other errors.

C_L, CV and GCV can all be viewed as adjustments to the squared residual which correct for $2\sigma^2 w_\lambda(x_{n,i}, i)$. Mallow's C_L is defined by

$$r^2_{C_L}(x_{n,i}, \lambda, n) = r^2(x_{n,i}, \lambda, n) + 2\hat{\sigma}^2 w_\lambda(x_{n,i}, i)$$

where $\hat{\sigma}^2$ is some unbiased estimator of σ^2. (In Mallows' original paper, the criterion is divided by $\hat{\sigma}^2$.) For bandwidth selection, the criterion is usually totalled over all the design points and the value of the smoothing parameter which minimizes this sum is selected. However, the theoretical computations in this section are done pointwise.

When the errors are independent, $V_2(x_{n,i}, \lambda, n) = 0$, and C_L is unbiased for ESPE. However, $r^2_{C_L}(x_{n,i}, \lambda, n)$ can be badly biased for $ESPE(x_{n,i}, \lambda, n)$ if $V_2(x_{n,i}, \lambda, n)$ is large.

Mallow's C_L is inconvenient to use for bandwidth selection, particularly when the errors are correlated, because of the difficulty in finding good estimators of σ^2. CV and GCV are adjustments to the residual sum of squares which have

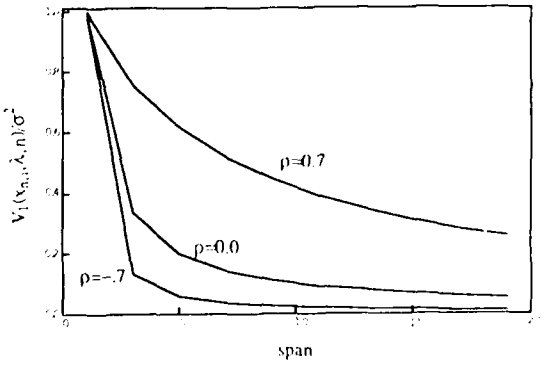


Figure 4: *Variance term, $V_1(x_{n,i}, \lambda, n)/\sigma^2$, for the uniform kernel with AR(1) errors, various values of $\rho(1)$.*

the same asymptotic expectation as r^2_{CL} and do not require an estimate of the variance.

Algebraic manipulation shows that the CV criterion is

$$r^2_{CV}(x_{n,i},\lambda,n) = \frac{r^2(x_{n,i},\lambda,n)}{(1-w_\lambda(x_{n,i},i))^2}.$$

GCV was proposed by Craven and Wahba (1978) as an adjustment to cross-validation that is more nearly unbiased for ESPE in the case of unequally spaced points, if the design points are considered to be fixed. The GCV criterion is

$$r^2_{GCV}(x_{n,i},\lambda,n) = \frac{r^2(x_{n,i},\lambda,n)}{\left(1-\frac{1}{n}trW_{\lambda,n}\right)^2}$$

where $W_{\lambda,n}$ is the matrix $[w_\lambda(x_{n,i},j)]$ and $trW_{\lambda,n} = \sum_{i=0}^{n} w_\lambda(x_{n,i},i)$. If λ is small, $\frac{trW_{\lambda,n}}{n} \approx w_\lambda(x_{n,i},i) \approx \frac{K(0)}{n\lambda}$ so CV and GCV differ very little.

For $\lambda \to 0$ and $n\lambda \to \infty$, $MSE(x_{n,i},\lambda,n) = O(\frac{1}{n\lambda}) + O(\lambda^{2p})$, while Lemma 2. below, shows that $V_2(x_{n,i},\lambda,n) = 0(\frac{1}{n\lambda})$. Using a Taylor series expansion for CV or GCV, the expectation of the criteria are:

$$E(r^2_{(G)CV}(x_{n,i},\lambda,n) = \sigma^2 + MSE(x_{n,i},\lambda,n) - V_2(x_{n,i},\lambda,n) + o(\lambda^{2p}) + o(\frac{1}{n\lambda}). \quad (5)$$

So, asymptotically, C_L , CV, and GCV have the same expectation for equally spaced design points. Theorem 2 describes the behavior of this expectation.

Lemma 2: *If the kernel function satisfies conditions A-C, and the correlation function satisfies condition D, then for* $\frac{\lambda}{2} < x < 1 - \frac{\lambda}{2}$,

$$V_2(x_{n,i},\lambda,n) = 4\sigma^2\frac{K(0)}{n\lambda}S_\rho + o(\frac{1}{n\lambda}).$$

Theorem 2: *Under the conditions of Theorem 1, the variance term for* C_L , *CV, and GCV is*

$$V_1(x_{n,i},\lambda,n) - V_2(x_{n,i},\lambda,n) = \sigma^2(1 + 2S_\rho(1 - 2\frac{K(0)}{W_K}))$$

The details of the proof of Lemma 2 are in Altman, 1988. Theorem 2 follows simply. Theorem 2 is also true for suitable boundary kernels.

Most kernel estimators commonly in use have $K(0) \geq K(x)$, so that $K(0) \geq W_K$. Let $\lambda^*(y)$ be the bandwidth chosen by one of the selection criteria for estimating f from y, and $\lambda^*(Z)$ be the bandwidth chosen when estimating from Z of Corollary 1. (Recall that Z is a process with the same mean and variance as y and independent errors.) Theorem 2 suggests that if $S_\rho < 0$, then $\lambda(y) < \lambda(Z) \approx \lambda^*(Z) < \lambda^*(y)$. If $S_\rho > 0$,then $\lambda(y) > \lambda(Z) \approx \lambda^*(Z) > \lambda^*(y)$. In fact, if $2S_\rho(1 - 2\frac{K(0)}{W_K}) < 0$, then the criteria tend to be ***strictly increasing*** with λ, and so they will favor interpolation. This is supported by the simulation results reported in Altman 1987 and 1988.

4. Correcting for Correlation

Theorem 2 establishes that bandwidth selectors perform poorly because they do not fully correct the residual sum of squares. In this section, two methods are suggested for correcting the selection criteria when the correlation function is known. The direct method adjusts the ***criteria*** to make them more nearly unbiased for ESPE. The indirect method transforms the ***residuals*** to produce transformed residuals which are less correlated.

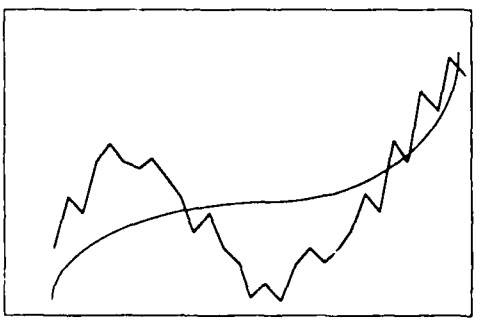
If the correlation function ρ_n is known, with corresponding correlation matrix R_n, Mallow's C_L can be corrected to be an unbiased estimator of ESPE.

From equation (4) an appropriate adjustment for Mallow's C_L criterion is

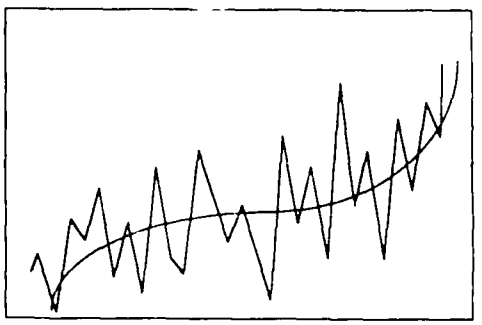
$$r^2_{CL,\rho}(x_{n,i},\lambda,n) = r^2(x_{n,i},\lambda,n) + 2\sigma^2\sum_{j=-k}^{k} w_\lambda(x_{n,i},i+j)\rho_n(j). \quad (6)$$

The corresponding adjustments for CV and GCV are intended to match the low order terms in the Taylor series expansion in equation (5) to the adjusted C_L criterion in equation (6). One way to do this is to set

$$r^2_{CV,\rho}(x_{n,i},\lambda,n) = \frac{r^2(x_{n,i},\lambda,n)}{(1-\sum_{j=-k}^{k} w_\lambda(x_{n,i},i+j)\rho_n(j))^2}$$



Positively correlated errors



Negatively correlated errors

Figure 5: *Positively correlated errors require large bandwidths to average to zero, while negatively correlated errors require only small ones.*

and

$$r^2_{GCV,\rho}(x_{n,i},\lambda,n) = \frac{r^2(x_{n,i},\lambda,n)}{(1-\frac{1}{n}\mathrm{tr}W_{\lambda,n}R_n)^2}$$

We will call this the direct method of correcting for correlation, and denote the corresponding bandwidth selection criteria by CV_ρ and GCV_ρ respectively.

Another approach to the problem when the correlation matrix is known, is to compute the transformed residuals: $r_{\rho^{-1}}(\bullet,\lambda,n) = R_n^{-\frac{1}{2}} r(\bullet,\lambda,n)$. This has been used with some success in the context of spline smoothing with normal AR(1) errors, (Diggle 1985, Diggle and Hutchinson, 1985, and Engle et al., 1986). The goodness of fit criterion is then the total weighted MSE,

$$\begin{aligned} TSE_{\rho^{-1}}(\lambda,n) &= E(\hat{f}_{\lambda,n}(\bullet) - f(\bullet))' R_n^{-1} (\hat{f}_{\lambda,n}(\bullet) - f(\bullet)) \\ &= \mathrm{tr}B(\bullet,\lambda,n)B(\bullet,\lambda,n)R_n^{-1} + \sigma^2 \mathrm{tr}W'_{\lambda,n}W_{\lambda,n} \end{aligned}$$

where $B(\bullet,\lambda,n)$ is the vector of biases defined by (3).

The totalled C_L criterion based on the transformed residuals,

$$\sum r^2_{CL,\rho^{-1}} = \sum_{i=0}^{n} r^2_{\rho^{-1}}(x_{n,i},\lambda,n) + 2\hat{\sigma}^2 \ \mathrm{tr}\ W_{\lambda,n},$$

is then unbiased for the expected value of $TSE_{\rho^{-1}}(\lambda,n)$.

The CV and GCV criteria based on the transformed residuals can also be readily defined. They are

$$r^2_{CV,\rho^{-1}}(\lambda,n) = \frac{r_{\rho^{-1}}(x_{n,i},\lambda,n)}{(1-w_\lambda(x_{n,i},i))^2}.$$

and

$$r^2_{GCV,\rho^{-1}}(\lambda,n) = \frac{r^2_{\rho^{-1}}(x_{n,i},\lambda,n)}{(1-\frac{1}{n}\ \mathrm{tr}\ W_{\lambda,n})^2}.$$

We will call this the indirect method of correcting for correlation, and denote the corresponding bandwidth selection criteria by $\sum CV_{\rho^{-1}}$ and $\sum GCV_{\rho^{-1}}$ respectively.

Simulation results reported in Altman 1987 and 1988 show that both criteria perform well when the correlation function is known.

5. Estimating the Correlation Function

Usually the correlation function is unknown and must be estimated from the data. Theorem 3 below, shows that the method of moments (MM) estimator of $\rho_n(s)$ is consistent under mild regularity conditions on the errors. The corollaries explore the nature of the bias for finite samples.

Theorem 3: ***Suppose the mean function has p^{th} derivative which is Lipschitz of order γ, and the kernel and correlation functions satisfy the conditions of Theorem 1. For fixed s, define the method of moments estimator of $\rho_n(s)$ by***

$$\hat{\rho}_n(s,\lambda) = \frac{\sum_{i=[\frac{n\lambda}{2}]}^{n+1-[\frac{n\lambda}{2}]-s} r(x_{n,i},\lambda,n)r(x_{n,i+s},\lambda,n)}{\sum_{i=[\frac{n\lambda}{2}]}^{n+1-[\frac{n\lambda}{2}]-s} r^2(x_{n,i},\lambda,n)}.$$

Then, as $\lambda \to 0$ and $n\lambda \to \infty$

$$E(\hat{\rho}_n(s,\lambda)) = \frac{\rho_n(s) + C(\lambda,n)}{1+C(\lambda,n)} + o(\lambda^{2p}) + o(\frac{1}{n\lambda}).$$

where

$$\begin{aligned} C(\lambda,n) = 1 + \lambda^{2p}\left(\frac{s_K}{p!}\right)^2 \frac{\int (f^{(p)}(x))^2 dx}{\sigma^2} \\ + \frac{(1+2S_\rho)}{n\lambda}(W_K - 2K(0)). \end{aligned} \tag{7}$$

Suppose in addition, the errors are fourth order stationary. Let $\kappa_{4,n}(r,s,0)$ be the fourth joint cumulant of the distribution of $(\varepsilon_{n,t},\varepsilon_{n,t+r},\varepsilon_{n,t+s},\varepsilon_{n,t+r+s})$ and assume that, for n sufficiently large, and for all r and s, $\sum_{r=-\infty}^{\infty} |\kappa_{4,n}(r,s,0)| < \infty$. Then

$$Var(\hat{\rho}_n(1,\lambda)) = O(\frac{1}{n\lambda})$$

In any finite sample, the estimator is biased. Since the information about the correlation is in the errors, it is not surprising that the bias is least when the signal to noise ratio is small. When the signal to noise ratio is large, bandwidth selection can readily be done by eye. Estimates of the correlations are best when they are most needed.

Corollary 3.1: *Under the conditions of Theorem 3, asymptotically,*

a) The bias of $\hat{\rho}_n(1,\lambda)$ is a function of the signal to noise ratio, $\int (f^{(p)}(x))^2 dx / \sigma^2$.

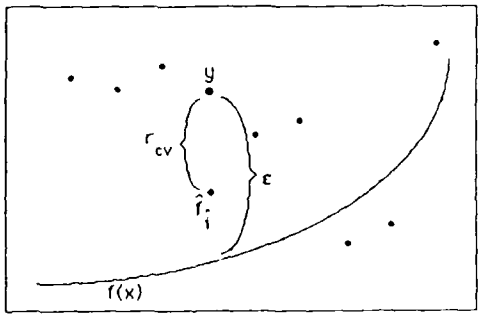


Figure 6: *When the errors are positively correlated, the cross-validation estimator lies too close to the data. As a result, the "deleted" residuals are too small, and CV is biased down*

b) If $2K(0) < W_K$, the bias of $\hat{\rho}_n(1,\lambda)$ is positive.
c) If $2K(0) \geq W_K$, $\hat{\rho}_n(1,\lambda)$ has bias which is increasing in λ, and is decreasing in S_ρ.

Proof:

$$E(\hat{\rho}_n(1,\lambda)) \approx \frac{\rho_n(1) + C(\lambda,n)}{1 + C(\lambda,n)}$$

where $C(\lambda,n)$ is defined by (7). $E(\hat{\rho}_n(1,\lambda))$ is an increasing function of $C(\lambda,n)$. If $2K(0) < W_K$, then $C(\lambda,n) > 0$. If $2K(0) \geq W_K$, then $C(\lambda,n)$ is an increasing function of λ, and of the signal to noise ratio and a decreasing function of S_ρ.

From these computations, it is possible to compute the bandwidth which is asymptotically optimal for minimizing the bias of the MM estimator. For kernels which have a maximum at $K(0)$, this bandwidth is not the asymptotically optimal bandwidth for estimating the mean function. This leads to the tentative conclusion that techniques which iteratively compute the mean and correlation function may not converge to the true mean and correlation function. However, bimodal kernels may have some promise for this situation.

In practice, choice of bandwidth does not appear to be very sensitive to the estimated correlation (Altman, 1987 and 1988, Engle et al, 1985). In consequence, a simple two step procedure often performs well. First, estimate the correlations from the residuals from a moderate bandwidth smooth. Then use the estimated correlations to pick a bandwidth for estimating the mean function f.

For comparison purposes, Table 1 shows an example of *the results from the simulation study in Altman*, 1987. Fifty realizations were taken of a sample of size 128 from the process $y = \cos(3.15\pi x) + \epsilon$ where the the errors ϵ was a Normal AR(1) process with variance 1.0. The quadratic kernel, $K(x) = 3(\frac{1}{4} - x^2)$, was used. The number under the heading "min ASE" is the median over the 50 realizations of the average squared error (ASE) loss at the bandwidth minimizing ASE for that realization. The numbers under the heading "min CV" is the median of the ratio of ASE at the bandwidth selected by CV to the minimum ASE for that realization. The numbers under "min CV_ρ" and "min $CV_{\hat{\rho}}$" are similar ratios for the direct correction to CV with the true and estimated values of the correlation function.

Table 1

$\rho(1)$	min ASE	min CV	min CV_ρ	min $CV_{\hat{\rho}}$
.9	.435	2.03	1.12	1.11
.6	.108	1.49	1.12	1.07
0	.051	1.06	1.06	1.11
-.9	.0047	1.30	2.13	1.17

The results were very similar for all the kernels used in the study. Results were also very good when the signal to noise ratio was very large (error variance .01) even though the estimates of correlation showed a very pronounced upwards bias.

Acknowledgements: *This work was supported by a Natural Sciences and Engineering Research Council of Canada postgraduate scholarship, and Hatch Grant 151410 NYF. Much of this work was completed while the author was in the Department of Statistics at Stanford University. The author would like to thank Iain Johnstone for his valuable guidance in supervising the dissertation from which this article was taken. Discussions with Brad Efron, Jerome Friedman, and Peter Lewis also contributed substantially to the ideas in this article.*

References

Allen, D.M. (1974) The relationship between variable selection and data augmentation and a method for prediction. *Technometrics* **16** 1307-1325.

Altman, N. S. (1987) Smoothing Data with Correlated Errors. Stanford Department of Statistics Technical Report No. 280.

Altman, N. S. (1988) Kernel Smoothing of Data with Correlated Errors. Cornell Biometrics Unit Report BU-981-M.

Bartlett, M. S. (1946) On the theoretical specification and sampling properties of autocorrelated time series *Suppl. J. Roy. Statist. Soc.* **8** 27-41

Becker, R.A., Chambers, J.M. (1984) **S An Interactive Environment for Data Analysis and Graphics** Wadsworth Statistics/Probability Series.

Benedetti, J.K. (1977) On the Nonparametric Estimation of Regression Functions. *J.R. Statist. Soc.* B **39** 248-253.

Box, G.E.P. and Jenkins, G.M. (1976) **Time Series Analysis: Forecasting and Control**. Holden-Day, Inc. San Francisco.

Collomb G. *(1985) Nonparametric Regression. An Up-to-*Date Bibliography *Statistics* **16** 309-324.

Craven, P. and Wahba, G. (1979) Smoothing Noisy Data with Spline Functions. *Numer. Math.* **31** 377-403.

Diggle, P.J. (1985) Discussion of "Some Aspects of the Spline Smoothing Approach to Non-parametric Regression Curve Fitting" by B.W. Silverman. *J.R. Statist. Soc. B* **47** 1-52.

Diggle, P.J. and Hutchinson, M.F. (1985) Spline smoothing with autocorrelated errors. CSIRO Technical Report.

Eagleson, G.K. and Buckley, M.J. (1987) Estimating the Variance in Non-Parametric Regression. C.S.I.R.O., Division of Mathematics and Statistics, West Lindfield, Australia (preprint).

Eagleson, G.K. Silverman, B.W. Buckley, M.J. (1987) The Estimation of Residual Variance in Non-parametric Regression (preprint).

Efron, B. (1986) How Biased is the Apparent Error Rate of a Prediction Rule? *JASA* **81** 461-470.

Engle, R. Granger C.W.J. Rice, J. Weiss A. (1986) Semiparametric Estimates of the Relation Between Weather and Electricity Sales *JASA* **81** 310-320.

Franke, R. The Performance of Generalized Cross Validation with Laplacian Smoothing Splines. Naval Postgraduate School, Monterey, California Technical Report NPS-53-85-008S.

Friedman, J.H. (1984) A Variable Span Smoother. LCS 5, Stanford, California.

Gasser, T., Müller, H-G. (1979) Kernel estimation of regression functions *Smoothing Techniques for Curve Estimation* 23-67 **Lecture Notes in Math.** 757. Springer-Verlag, Berlin.

Gasser, T., Müller, H-G., Köhler, W., Molinari, L. and Prader, A. (1984) Nonparametric Regression Analysis of Growth Curves. *Ann. of Stat.* **12** 210-229.

Geisser, S. (1975) The predictive sample reuse method with applications *JASA* **70** 320-328.

Greblicki, W., Krzyzak, A., Pawlak, M. (1984) Distribution-Free Pointwise Consistency of Kernel Regression Estimate. *Ann. of Stat.* **12** 1570-1575.

Härdle, W. and Kelly, G. (1985) Nonparametric Kernel Regression Estimation - Optimal Choice of Bandwidth. Division of Biostatistics, Stanford University Technical Report 100.

Härdle, W. Hall, P., Marron, J.S. (1988) How Far are Automatically Chosen Regression Smoothing Parameters from their Optimum? *JASA* **83** 86-95.

Härdle,W. and Marron, J.S. (1985) Asymptotic nonequivalence of some bandwidth selectors in nonparametric regression. *Biometrika* **72** 481-484.

Hart, J.D. (1987) Kernel Regression with Time Series Errors (preprint)

Hart, J.D. and Wehrly, T.E. (1986) Kernel Regression Estimation Using Repeated Measurements Data. *JASA* **81** 1080-1088.

Huber P. J.(1985) Projection Pursuit. *Ann. of Stat.* **13** 435-475.

IMSL Incorporated (1984) **IMSL Library Reference Library** Houston.

Li, K-C. (1985) Consistency for Cross-Validated Nearest Neighbor Estimates in Nonparametric Regression. *Ann. of Stat.* **12** 230-240.

Li, K-C. (1985) From Stein's Unbiased Risk Estimates to the Method of Generalized Cross Validation. *Ann. of Stat.* **13** 1352-1377.

Li, K-C. (1986) Asymptotic Optimality of C_L and Generalized Cross-Validation in Ridge Regression with Application to Spline Smoothing. *Ann. of Stat.* **14** 1101-1112.

Mallows, C.L. (1973) Some Comments on C_p. *Technometrics* **15** 661-675.

McDonald, J.A. (1983) Periodic Smoothing of Time Series. Project ORION, Stanford University Technical Report 017.

Nadaraya, E.A. (1964) On estimating regression. *Theory of Probability and its Applications* **9** 141-142.

Parzen, E. (1959) Statistical Inference on Time Series by Hilbert Space Methods, I. Department of Statistics, Stanford University Technical Report No. 23 (NR-042-993).

Parzen, E. (1961) Regression Analysis of Continuous Parameter Time Series. Fourth Berkeley Symposium.

Priestly, M.B. and Chao, M.T. (1972) Non-parametric function fitting. *J.R. Statist. Soc.* B **34** 385-392.

Reinsch, C.H. (1967) Smoothing by Spline Functions. *Numerische Mathematik* **10** 177-183.

Rice, J. (1984) Bandwidth Choice for Nonparametric Regression. *Ann. of Stat.* **12** 1215-1230.

Rice, J. and Rosenblatt, M. (1983) Smoothing Splines: Regression, Derivatives and Deconvolution. *Ann. of Stat.* **11** 141-156.

Shibata, R. (1981) An optimal selection of regression variables. *Biometrika* **68** 45-54.

Silverman, B.W. (1984) Spline Smoothing: The Equivalent Variable Kernel Method. *Ann. of Stat.* **12** 898-916.

Silverman, B.W. (1985) Some Aspects of the Spline Smoothing Approach to Non-parametric Regression Curve Fitting. *J.R. Statist. Soc. B* **47** 1-52.

Stone, C.J. (1977) Consistent Nonparametric Regression. *Ann. of Stat.* **5** 595-645.

Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions. *J.R. Statist. Soc. B* **36** 111-147.

Stone, M. (1977) An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J.R. Statist. Soc. B* **39** 44-47.

Stute, W. (1984) Asymptotic Normality of Nearest Neighbor Regression Function Estimates. *Ann. of Stat.* **12** 917-926.

Wahba, G. (1977) Practical Approximate Solutions to Linear Operator Equations when the Data are Noisy. *SIAM J. Numer. Anal.* **14** 651-667.

Wahba, G. (1978) Improper Priors, Spline Smoothing and the Problem of Guarding Against Model Errors in Regression. *J.R. Statist. Soc. B* **40** 364-372.

Wahba, G. (1983) Bayesian "Confidence Intervals" for the Cross-validated Smoothing Spline. *J.R. Statist. Soc. B* **45** 133-150.

Wahba, G. (1984) Cross-Validated Spline Methods for the Estimation of Multivariate Functions from Data on Functionals. *Proceedings 50th Anniversary Conference Iowa State Statistical Laboratory*, H.A. David and H.T. David, Editors, The Iowa State University Press.

Watson, G.S. (1964) Smooth Regression Analysis. *Sankhyā, Series A* **26** 359-372.

Whittaker, E. (1923) On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society* **41** 63-75.

Yang, S-S. (1981) Linear Functions of Concomitants of Order Statistics with Application to Nonparametric Estimation of a Regression Function. *JASA* **76** 658-662.

DERIVATIVE ESTIMATION BY POLYNOMIAL–TRIGONOMETRIC REGRESSION

Randy Eubank, Southern Methodist University and Paul Speckman, University of Missouri

ABSTRACT

Let μ be a smooth function defined on an interval [a,b], and suppose that $y_1,\cdots,y_n$ are uncorrelated observations with $E(y_j) = \mu(t_j)$, $1 \le j \le n$, where the t_j are fixed equally spaced points in [a,b]. Estimation of μ and its derivatives by regression on trigonometric and low order polynomial terms is considered. The polynomial terms are shown to adjust for the boundary bias problems known to be suffered by regression on trigonometric terms alone. As a result, the estimator of μ and its derivatives obtained by this method is shown to be competitive with other nonparametric estimators. The method is illustrated by estimating two growth curves and their derivatives.

1. INTRODUCTION

A problem that arises in nonparametric regression analysis is the estimation of functionals of a regression curve. One particularly important example is estimating a derivative of some order. Thus in this paper we investigate the properties of a simple new method for derivative estimation. It will be shown that derivative estimation by regression on a combination of polynomial and trigonometric functions provides optimal rates of convergence with respect to average mean square error. These results extend work on function estimation by Eubank and Speckman (1988) to derivative estimation under the assumption of equally spaced observations.

Assume that observations are taken according to the model

$$y_i = \mu(t_i) + \epsilon_i, \quad 1 \le i \le n,$$

where the ϵ_i are zero mean uncorrelated random variables with constant variance σ^2. Further assume that the t_i are equally spaced, and without loss of generality take $t_i = (i-1)/n$, $1 \le i \le n$. An estimator for μ that was proposed by Eubank and Speckman (1988) is

$$\mu_\lambda(t) = b_0 + \sum_{j=1}^{d} b_j t^j + \sum_{k=1}^{\lambda} (c_k \cos 2\pi kt + s_k \sin 2\pi kt), \tag{1.1}$$

where the b_j, c_k, and s_k are obtained by minimizing over B_j, C_k, and S_k the quantity

$$\sum_{i=1}^{n} [y_i - B_0 - \sum_{j=1}^{d} B_j t_i^j - \sum_{k=1}^{\lambda} (C_k \cos 2\pi kt_i + S_k \sin 2\pi kt_i)]^2.$$

Here the terms d and λ are "smoothing parameters" to be chosen by the user. A workable strategy is to fix d at a small value (typically 2 or 3) and allow λ to vary with n to obtain a suitable fit (cf. Eubank and Speckman (1988)).

A natural estimator of the mth derivative of μ can be obtained by differentiating (1.1) m times. We will label the result $\mu_\lambda^{(m)}(t)$ and term it a polynomial–trigonometric regression (PTR) estimator of $\mu^{(m)}(t)$. This is the estimator to be studied in the present paper.

The proposed method is motivated by the following observations. It is known (see Eubank, Hart, and Speckman (1987), for example) that estimation by a trigonometric series alone (i.e. with no polynomial part in (1.1)) has optimal convergence properties if μ has d derivatives with $\mu^{(m)}(0) = \mu^{(m)}(1)$, $0 \le m < d$. The problem with the pure trigonometric series is the fact that the fit is necessarily periodic while the true response function μ need not be. This can result in serious bias problems at the boundaries. However, suppose that $p(t)$ is the unique polynomial of degree d such that

$$p^{(m)}(1) - p^{(m)}(0) = \mu^{(m)}(1) - \mu^{(m)}(0), \quad 0 \le m < d, \tag{1.2}$$

and let $\mu(t) = p(t) + \mu_0(t)$. Then μ_0 has the requisite boundary properties for good estimation by a trigonometric series. Heuristically, the polynomial part of PTR estimates p, and the trigonometric part effectively models μ_0. We make these observations precise in the next sections.

The performance of the derivative estimator obtained from (1.1) will be measured by the average mean square error of estimation,

$$AMSE(\lambda) = E\{n^{-1} \sum_{i=1}^{n} [\mu^{(m)}(t_i) - \mu_\lambda^{(m)}(t_i)]^2\}.$$

Our principle result about the properties of $\mu_\lambda^{(m)}$ is the following.

Theorem. If μ has $d-1$ absolutely continuous derivatives with $\mu^{(d)} \in L^2[0,1]$ and $0 \le m < d$, then as $n \to \infty$ and $\lambda \to \infty$ in such a way that $\lambda^2/n \to 0$,

$$AMSE(\lambda) = O(\lambda^{-2(d-m)}) + O(\lambda^{2m+1}/n).$$

In particular, for $\lambda = O(n^{1/(2d+1)})$,

$$AMSE(\lambda) = O(n^{-2(d-m)/(2d+1)}).$$

Remark. When $\lambda = O(n^{1/(2d+1)})$, the resulting rate of decay of $AMSE(\lambda)$ is the "optimal" uniform rate of convergence for derivative estimation in the class of functions $\mu^{(d)} \in L_2[0,1]$ as shown by Stone (1982). This rate is achieved by a great many nonparametric estimators.

We believe that the assumption of equally spaced

points can be weakened substantially. In Eubank and Speckman (1988), we were able to obtain the results of the theorem for the case $m = 0$ assuming only that the t_j's were a sample from a distribution with positive bounded density on [0,1]. We have been able to extend the methods used there to obtain good bias bounds for derivative estimation, but at present we are unable to get a satisfactory estimate of the variance in the general case. However, we conjecture that the hypothesis of equally spaced points is not necessary for derivative estimation.

In the next section we discuss some preliminaries, establish further notation, and define a particular basis for the polynomial part of the regression. The proof of the main result is then sketched in Section 3. An example is presented in Section 4.

2. PRELIMINARIES

To begin, note that $\mu_\lambda(t)$ may be obtained by using OLS to fit

$$\mu_\lambda(t) = \sum_{j=1}^{d} b_j p_j(t) + \sum_{k=-\lambda}^{\lambda} a_k \exp\{2\pi i k t\}, \tag{2.1}$$

where $\{1, p_j,\ j=1,\cdots,d\}$ is any linearly independent system of polynomials spanning $\{1,\cdots,t^d\}$. This representation is useful because it is easier to work with analytically, and we will freely use the fact that the left hand side of (2.1) is real. A particularly convenient basis for the polynomial term is obtained by defining $p_j(t)$, $1 \le j \le d$, to be the *unique polynomial of degree* j satisfying

$$\int_0^1 p_j(t)dt = 0,$$
$$p_j^{(m)}(0) = p_j^{(m)}(1),\ 0 \le m \le j-2 \text{ for } j \ge 2,$$
$$p_j^{(j-1)}(1) - p_j^{(j-1)}(0) = 1.$$

It can be shown that $p_j(t) = B_j(t)/j!$, where $B_j(t)$ is the jth Bernoulli polynomial.

For simplicity, we will assume that n is odd. The case of n even is similar. Then the vectors $\mathbf{x}_k = (1,\exp\{2\pi ik/n\},\cdots,\exp\{2\pi ik(n-1)/n\})^t$, $k = 0, \pm 1,\cdots, \pm(n-1)/2$, form an orthogonal basis for $\mathbb{R}^n$. Hence $\mathbf{y} = \sum_{|k|\le(n-1)/2} a_k \mathbf{x}_k$, where $a_k = n^{-1}\sum_{r=1}^{n} y_r \exp\{-2\pi ik(r-1)/n\}$. If we let $\mathbf{T}_{n\lambda}\mathbf{y}$ denote the projection of $\mathbf{y}$ onto the span of $\{\mathbf{x}_k : |k| \le \lambda\}$, the orthogonality of the $\mathbf{x}_k$ implies that

$$\mathbf{T}_{n\lambda}\mathbf{y} = \sum_{|k|\le\lambda} a_k \mathbf{x}_k = n^{-1}\mathbf{X}_{1n}\mathbf{X}_{1n}^{*}\mathbf{y},$$

where $\mathbf{X}_{1n} = \{\exp(2\pi ikt_r)\}_{1\le r\le n,\ -\lambda\le k\le\lambda}$ is a complex valued $n \times (2\lambda+1)$ matrix. ($\mathbf{X}_{1n}^{*}$ denotes the complex conjugate transpose.) Now let $\mathbf{p}_j = (p_j(t_1),\cdots,p_j(t_n))^t$ and define $\mathbf{p}_{jn\lambda} = \lambda^{j-1/2}(\mathbf{I} - \mathbf{T}_{n\lambda})\mathbf{p}_j$. (The reason for the normalization by $\lambda^{j-1/2}$ will become clear in Section 3.) The $\mathbf{p}_{jn\lambda}$ are orthogonal to the $\mathbf{x}_k$. With $\mathbf{X}_{2n} = \{\mathbf{p}_{1n\lambda},\cdots,\mathbf{p}_{dn\lambda}\}$, the $n \times d$ matrix with columns $\mathbf{p}_{jn\lambda}$, it follows that $\mathbf{T}_{n\lambda}$ and $\mathbf{P}_{n\lambda} = \mathbf{X}_{2n}(\mathbf{X}_{2n}^{*}\mathbf{X}_{2n})^{-1}\mathbf{X}_{2n}^{*}$ are orthogonal, and the solution to the least squares problem giving (2.1) can be expressed as

$$\boldsymbol{\mu}_\lambda = (\mu_\lambda(t_1),\cdots,\mu_\lambda(t_n))^t = (\mathbf{T}_{n\lambda} + \mathbf{P}_{n\lambda})\mathbf{y}.$$

To examine the behavior of the derivative estimate, we need a representation of the *function* $\mu_\lambda(t)$ defined on [0,1]. Let $\mathbf{a} = (a_{-\lambda},\cdots,a_\lambda)^t = n^{-1}\mathbf{X}_{1n}^{*}\mathbf{y}$, and define $\mathbf{T}_{n\lambda}\mathbf{y}(t) = \sum_{|k|\le\lambda} a_k \exp\{2\pi ikt\}$. For a function $f(t)$ on [0,1], the projection $\mathbf{T}_{n\lambda}f(t)$ will be defined by taking $\mathbf{y}^t = (f(t_1),\cdots,f(t_n))$. With this notation, $p_{jn\lambda}(t) = \lambda^{j-1/2}(1 - \mathbf{T}_{n\lambda})p_j(t)$. Setting $\mathbf{b} = (b_1,\cdots,b_d)^t = (\mathbf{X}_{2n}^{*}\mathbf{X}_{2n})^{-1}\mathbf{X}_{2n}^{*}\mathbf{y}$, we thus obtain

$$\mu_\lambda(t) = \sum_{|k|\le\lambda} a_k \exp\{2\pi ikt\} + \sum_{j=1}^{d} b_j p_{jn\lambda}(t).$$

Repeated differentiation then gives

$$\mu_\lambda^{(m)}(t) = \sum_{|k|\le\lambda} (2\pi ik)^m a_k \exp\{2\pi ikt\} + \sum_{j=1}^{d} b_j p_{jn\lambda}^{(m)}(t). \tag{2.2}$$

Now suppose p is the polynomial of degree d such that (1.2) holds. Because $1 - (\mathbf{T}_{n\lambda} + \mathbf{P}_{n\lambda})$ as an operator on $L_2[0,1]$ annihilates polynomials of degree d,

$$\mu(t) - E(\mu_\lambda(t)) = \mu(t) - (\mathbf{T}_{n\lambda} + \mathbf{P}_{n\lambda})\mu(t)$$
$$\mu_0(t) + p(t) - (\mathbf{T}_{n\lambda} + \mathbf{P}_{n\lambda})\mu_0(t) + p(t))$$
$$\mu_0(t) - (\mathbf{T}_{n\lambda} + \mathbf{P}_{n\lambda})\mu_0(t).$$

Differentiating this expression for the bias m times yields

$$\frac{d^m}{dt^m}\left\{\mu(t) - E(\mu_\lambda(t))\right\} = \mu_0^{(m)}(t) - \frac{d^m}{dt^m}(\mathbf{T}_{n\lambda} + \mathbf{P}_{n\lambda})\mu_0(t). \tag{2.3}$$

This representation shows that the behavior of the bias does not depend on the polynomial part, and the device of adding polynomial terms to the trigonometric regression frees the behavior of the bias from periodic boundary conditions.

Finally, to get a matrix representation for $\boldsymbol{\mu}_\lambda^{(m)} = (\mu_\lambda^{(m)}(t_1),\cdots,\mu_\lambda^{(m)}(t_n))^t$ and for the bias, let

$$\mathbf{Y}_{1n} = \{(2\pi ik)^m \exp(2\pi ikt_r)\}_{1\le r\le n;-\lambda\le k\le\lambda}$$

and

$$\mathbf{Y}_{2n} = \{p_{jn\lambda}^{(m)}(t_r)\}_{1\le r\le n;1\le j\le d}.$$

Equations (2.2) and (2.3) then become

$$\boldsymbol{\mu}_\lambda^{(m)} = n^{-1}\mathbf{Y}_{1n}\mathbf{X}_{1n}^{*}\mathbf{y} + \mathbf{Y}_{2n}(\mathbf{X}_{2n}^{*}\mathbf{X}_{2n})^{-1}\mathbf{X}_{2n}^{*}\mathbf{y} \tag{2.4}$$

and

$$\boldsymbol{\mu}^{(m)} - E(\boldsymbol{\mu}_\lambda^{(m)}) = \boldsymbol{\mu}_0^{(m)} - n^{-1}\mathbf{Y}_{1n}\mathbf{X}_{1n}^{*}\boldsymbol{\mu}_0 - \mathbf{Y}_{2n}(\mathbf{X}_{2n}^{*}\mathbf{X}_{2n})^{-1}\mathbf{X}_{2n}^{*}\boldsymbol{\mu}_0. \tag{2.5}$$

3. ASYMPTOTIC RESULTS

The proof of the theorem depends heavily on the Fourier series representation of μ_0. Notation and development here are adapted from Eubank (1988). Let α_k be the kth Fourier coefficient of μ_0 defined as

$$\alpha_k = \int_0^1 \exp\{-2\pi ikt\}\mu_0(t)dt.$$

Then μ_0 has the series representation

$$\mu_0(t) = \sum_{-\infty}^{\infty} \alpha_k \exp\{2\pi ikt\}.$$

Both sides of this expression can be differentiated (formally) m times to obtain the series expansion

$$\mu_0^{(m)}(t) = \sum_{-\infty}^{\infty} (2\pi ik)^m \alpha_k \exp\{2\pi ikt\}. \tag{3.1}$$

This shows that the kth Fourier coefficient of $\mu_0^{(m)}$ is $(2\pi ik)^m\alpha_k$. For $0 \le m < d$, the assumption $\mu_0^{(d)} \in L_2$ implies pointwise convergence in (3.1) (except at 0 and 1). Parseval's equality gives

$$\int_0^1 \mu_0^{(m)}(t)^2 dt = \sum_{-\infty}^{\infty} (2\pi k)^{2m}|\alpha_k|^2, \quad 0 < m < d. \tag{3.2}$$

The kth Fourier coefficient for μ_0 in R^n is defined to be

$$\alpha_{kn} = n^{-1}\sum_{r=0}^{n-1} \mu_0(r/n)\exp\{-2\pi ikr/n\}. \tag{3.3}$$

It can be shown (see Eubank (1988)) that

$$\bar{\alpha}_{kn} = \sum_{s=-\infty}^{\infty} \alpha_{k+ns}.$$

The proof of the main theorem rests on the following estimate for the convergence of the discrete Fourier coefficient $\bar{\alpha}_{kn}$. The proofs will be detailed elsewhere.

Lemma 1. Under the assumptions of the Theorem,

$$\bar{\alpha}_{kn} = \alpha_k + O(n^{-d}) \text{ for } |k| \le n/2. \tag{3.4}$$

The "big oh" term holds uniformly in n over the specified range of k.

Lemma 2. If $\lambda^2/n \to 0$,

$$n^{-1}\mathbf{p}_{un\lambda}^t\mathbf{p}_{vn\lambda} \to \begin{cases} 2(u+v-1)(-1)^{(u+3v)/2}, & u+v \text{ even}, \\ 0, & u+v \text{ odd}, \end{cases} \tag{3.5a}$$

and

$$n^{-1}\|\mathbf{p}_{jn\lambda}^{(m)}\|^2 = O(\lambda^{2m}), \quad 1 \le j < d. \tag{3.5b}$$

To proceed to the proof of the main theorem, write

$$AMSE(\lambda) = n^{-1}\sum_{i=1}^{n}[\mu^{(m)}(t_i) - E\mu_\lambda^{(m)}(t_i)]^2 + n^{-1}\sum_{i=1}^{n}\mathrm{Var}(\mu_\lambda^{(m)}(t_i)).$$

From (2.5), we can decompose the bias into two components, $\mathbf{b}_{1\lambda} = \boldsymbol{\mu}_0^{(m)} - n^{-1}\mathbf{Y}_{1n}\mathbf{X}_{1n}^{*}\boldsymbol{\mu}_0$ and $\mathbf{b}_{2\lambda} = -\mathbf{Y}_{2n}(\mathbf{X}_{2n}^{*}\mathbf{X}_{2n})^{-1}\mathbf{X}_{2n}^{*}\boldsymbol{\mu}_0$. The summed squared bias in AMSE is then $\|\mathbf{b}_{1\lambda}\|^2 + \|\mathbf{b}_{2\lambda}\|^2 + 2\mathbf{b}_{1\lambda}^t\mathbf{b}_{2\lambda}$. To obtain the desired rate of convergence for the bias, it suffices to show that $\|\mathbf{b}_{1\lambda}\|^2$ and $\|\mathbf{b}_{2\lambda}\|^2$ are both $O(n\lambda^{-2(d-m)})$.

We begin with $\mathbf{b}_{1\lambda} = (b_{1\lambda}(t_1),\cdots,b_{1\lambda}(t_n))^t$. By (3.3) and the orthogonality of the $\{\exp\{2\pi ikr/n\}\}$, $\Gamma_{n\lambda}\mu_0(t) = \sum_{|k|\le\lambda}\bar{\alpha}_{kn}\exp\{2\pi ikt\}$. Thus with $\bar{\alpha}_{kn}^{(m)} = n^{-1}\sum_{r=0}^{n-1}\mu_0^{(m)}(r/n)\exp\{-2\pi ikr/n\}$, we obtain for $t \in \{t_1,\cdots,t_n\}$

$$b_{1\lambda}(t) = \sum_{|k|\le(n-1)/2}\bar{\alpha}_{kn}^{(m)}\exp\{2\pi ikt\} - \sum_{|k|\le\lambda}(2\pi ik)^m\alpha_{kn}\exp\{2\pi ikt\}$$

and

$$n^{-1}\|\mathbf{b}_{1\lambda}\|^2 \le \sum_{|k|\le\lambda}|\bar{\alpha}_{kn}^{(m)} - (2\pi ik)^m\alpha_{kn}|^2 + \sum_{|k|>\lambda}|\bar{\alpha}_{kn}^{(m)}|^2. \tag{3.6}$$

The notation $\sum^{*}$ in the last line above denotes summation over the range $\lambda < |k| \le (n-1)/2$. From (3.4) applied to $\mu_0^{(m)}$ and the fact that $\mu_0^{(m)}$ has kth Fourier coefficient $(2\pi ik)^m \alpha_k$, we have $\bar{\alpha}_{kn}^{(m)} = (2\pi ik)^m \alpha_k + O(n^{-(d-m)})$. Hence using (3.4) in (3.6), we obtain the bounds

$$\begin{aligned} n^{-1}\|\mathbf{b}_{1\lambda}\|^2 &= \sum_{|k|\le\lambda} |(2\pi ik)^m \alpha_k + O(n^{-(d-m)}) \\ &\quad - (2\pi ik)^m (\alpha_k + O(n^{-d}))|^2 \\ &\quad + \sum^{*} |(2\pi ik)^m \alpha_k + O(n^{-(d-m)})|^2. \end{aligned}$$

The first sum on the right is bounded by $O(\lambda n^{-2(d-m)}) + O(\lambda^{2m+1} n^{-2d}) = o(\lambda^{-2(d-m)})$ as $\lambda^2/n \to 0$. The second sum is bounded by

$$\begin{aligned} &2\sum_{\lambda\le|k|} (2\pi k)^{2m} |\alpha_k|^2 + O(n^{-2(d-m)+1}) \\ &\quad \le 2(2\pi\lambda)^{-2(d-m)} \sum_{\lambda\le|k|} (2\pi k)^{2d} |\alpha_k|^2 \\ &\qquad + O(n^{-2(d-m)+1}) \\ &\quad = O(\lambda^{-2(d-m)}) + O(n^{-2(d-m)+1}) \end{aligned}$$

by (3.2). Using $\lambda^2/n \to 0$ again shows that

$$n^{-1}\|\mathbf{b}_{1\lambda}\|^2 = O(\lambda^{-2(d-m)}). \tag{3.7}$$

Next, equation (3.5a) implies that

$$n^{-1}\mathbf{X}_{2n}^{*}\mathbf{X}_{2n} \to \mathbf{G}, \tag{3.8}$$

where $\mathbf{G}$ is a $d \times d$ positive definite matrix. Using the L_2 matrix norm $\|\mathbf{A}\|^2 = \sup_{\|\mathbf{x}\|>0} \|\mathbf{A}\mathbf{x}\|/\|\mathbf{x}\|$ and the fact that $\|\mathbf{A}\|^2 \le \operatorname{tr} \mathbf{A}^{*}\mathbf{A}$, it can be shown that $n^{-1}\|\mathbf{b}_{2\lambda}\|^2$ is asymptotically bounded by $n^{-1}\operatorname{tr}(\mathbf{Y}_{2n}^{*}\mathbf{Y}_{2n})\|\mathbf{G}^{-1}\|^2(\operatorname{tr} \mathbf{G})(n^{-1}\|(\mathbf{I} - \mathbf{T}_{n\lambda})\boldsymbol{\mu}_0\|^2)$. But $n^{-1}\|(\mathbf{I} - \mathbf{T}_{n\lambda})\boldsymbol{\mu}_0\|^2 = O(\lambda^{-2d})$ by (3.7) with $m = 0$, and $n^{-1}\operatorname{tr}(\mathbf{Y}_{2n}^{*}\mathbf{Y}_{2n}) = O(\lambda^{2m})$ by (3.5b), hence $n^{-1}\|\mathbf{b}_{2\lambda}\|^2 = O(\lambda^{-2(d-m)})$. This completes the proof for the bias term.

To estimate the variance, recall that $\mathbf{X}_{1n}$ and $\mathbf{X}_{2n}$ are orthogonal and that $\mathbf{X}_{1n}^{*}\mathbf{X}_{1n} = n\mathbf{I}$. Then from (2.4) and (3.8),

$$\begin{aligned} \operatorname{tr} \operatorname{Var}(\boldsymbol{\mu}_\lambda^{(m)}) &\sim \sigma^2 n^{-1} \operatorname{tr}(\mathbf{Y}_{1n}^{*}\mathbf{Y}_{1n}) \\ &\quad + \sigma^2 \operatorname{tr}(\mathbf{Y}_{2n}^{*}\mathbf{Y}_{2n}(n\mathbf{G})^{-1}). \end{aligned}$$

The second term on the right is again $O(\lambda^{2m})$. For the first term, note that $n^{-1}\mathbf{Y}_{1n}^{*}\mathbf{Y}_{1n}$ has (u,v) element

$$\begin{aligned} &n^{-1}(-2\pi iu)^m (2\pi iv)^m \sum_{r=0}^{n-1} \exp\{2\pi i(u-v)r/n\} \\ &\quad = \begin{cases} 0 & u \ne v \\ (2\pi u)^{2m} & u = v. \end{cases} \end{aligned}$$

Consequently, $\operatorname{tr}(\mathbf{Y}_{1n}^{*}\mathbf{Y}_{1n}) = 2\sum_{k=1}^{\lambda} (2\pi k)^{2m} \sim 2(2\pi)^{2m}\lambda^{2m+1}/(2m+1)$. This completes the proof of the theorem.

4. A GROWTH CURVE APPLICATION

One application of nonparametric derivative estimation has been to the study of growth curves. The derivative of the growth curve, called velocity, is of special interest in analyzing growth spurts. An example using growth data supplied by Dr. L. Molinari on a boy and a girl is reported in Eubank (1988, pp. 156 ff., p. 186). Figure 1 shows plots of the raw data and the growth curve estimates using PTR. In this example, the parameters d and λ were both chosen using Generalized Cross Validation (GCV). The procedure, discussed in Eubank and Speckman (1988), is a data-based estimate of the parameters d and λ which would theoretically minimize AMSE. GCV selected $d = 3$ and $\lambda = 8$ for the boy and $d = 3$ and $\lambda = 4$ for the girl. The residuals for these fits are plotted in Figure 2.

The PTR derivative estimates are plotted in Figure 3. Because the method uses a projection, there may be a spurious local maximum in the estimate of the boy's velocity curve around 12 years. However growth spurts roughly at ages 7 and 14 are clearly visible and appear to be "real". This analysis agrees with the results from kernel smoothing reported in Eubank (1988). The velocity estimate for the girl is similar with two apparent growth spurts.

This analysis demonstrates the simplicity and usefulness of PTR. The PTR models can be fit with virtually any regression package. Thus good derivative estimates as in Figure 2 can be obtained with no specialized software. Because there are missing observations in both data sets, the assumption of equally spaced points does not hold, and the results of the theorem do not directly apply. However, we believe that the estimates obtained in these examples demonstrate that PTR can work in practice even for unequally spaced data.

REFERENCES

Eubank, R. (1988), *Spline Smoothing and Nonparametric Regression*, New York: Marcel Dekker.

Eubank, R. and Speckman, P. (1988), "Curve Fitting by Polynomial-Trigonometric Regression", manuscript.

Stone, C. (1982), Optimal Global Rates of Convergence for Nonparametric Regression," *Annals of Statistics*, 10, 1040–1053.

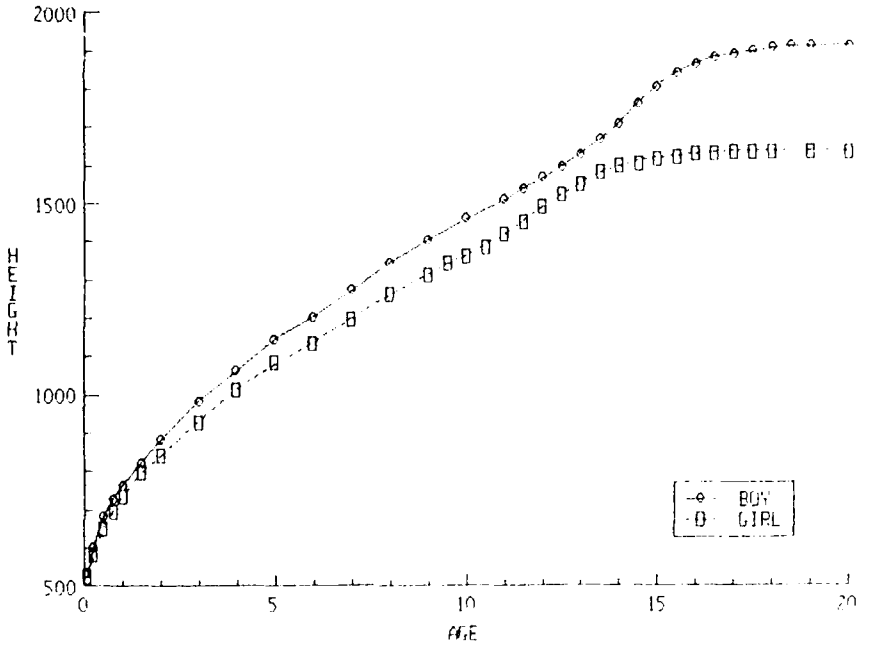


Figure 1. Height and estimated growth curves for a boy and a girl

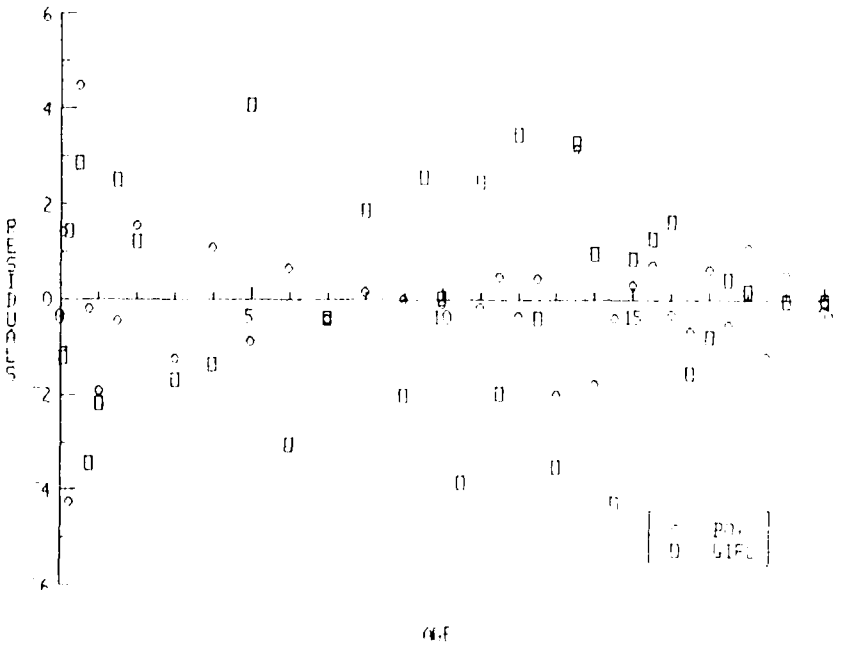


Figure 2. Residuals for estimated growth curves

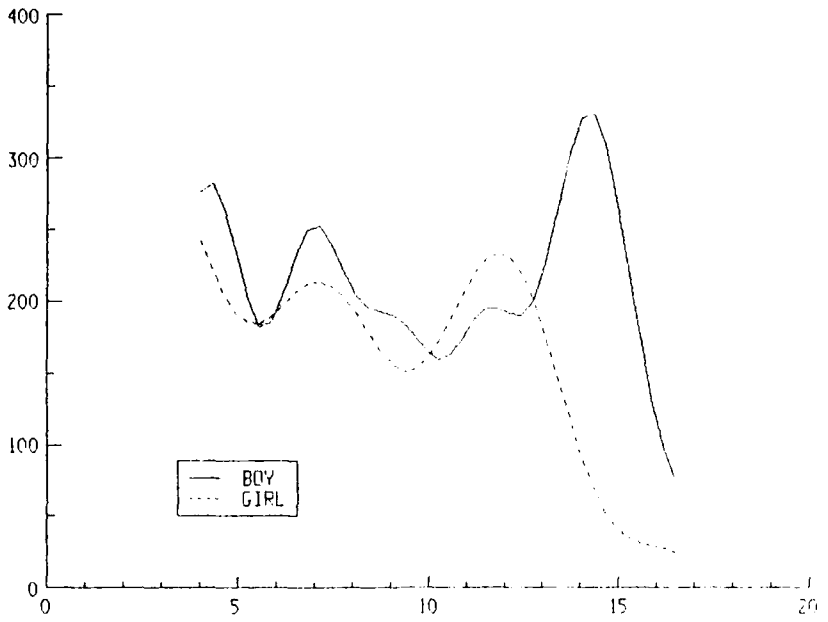


Figure 9. Estimated velocity curves for a boy and a girl

EFFICIENT ALGORITHMS FOR SMOOTHING SPLINE ESTIMATION OF FUNCTIONS WITH OR WITHOUT DISCONTINUITIES

Jyh–Jen Horng Shiau, University of Missouri–Columbia

ABSTRACT

Efficient algorithms are developed for GCV smoothing spline estimation of a function which is smooth except for some "break points" where discontinuities occur either in the function itself or its lower order derivatives. For a problem with n observations, these algorithms require $O(n)$ operations for the equally spaced knots case and $O(n^2)$ operations for the unequally spaced knots case. Similar algorithms are also derived for ordinary smoothing splines, that is, without discontinuities.

KEY WORDS: Smoothing splines, partial splines, discontinuities, efficient algorithms, generalized cross validation.

1. INTRODUCTION

To solve the problem of estimating an unknown function which is smooth except for break points (or curves or surfaces) where discontinuities occur either in the function itself or its lower order derivatives, Shiau (1985) proposed a partial spline approach to extend smoothing spline estimation method by augmenting it with jump functions to reflect the discontinuities. Shiau (1987) proposed some methods for inference on the magnitude of the jumps based on the mean square error of the estimate. Shiau, Wahba and Johnson (1986) employed the method to provide models to include specified discontinuities in otherwise smooth two or three dimensional objective analyses and demonstrated that the model is appropriate for including tropopause height information in temperature analysis. In this paper, we present some efficient algorithms for this particular problem in the univariate case which utilize a special structure of the covariance matrix of smoothing splines, and we show that the Generalized Cross Validation (GCV) method of choosing the smoothing parameter as well as the smoothing spline estimate can be achieved in $O(n)$ operations for equally spaced data and $O(n^2)$ operations for unequally spaced data, where n is the number of observations in the data.

Given noisy data $\{y_i, i=1,2,...,n\}$ observed from an unknown function g at $\{t_i, i=1,2,...,n\}$ in the interval $[0, 1]$, we consider the following nonparametric regression model:

$$y_i = g(t_i) + \epsilon_i, \; i = 1,2,...,n, \tag{1.1}$$

where ϵ_i's are uncorrelated with mean zero and common variance σ^2. For ordinary spline smoothing, the estimate g_λ is the minimizer of the following variational problem:

$$\min_{f \in H} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(t_i))^2 + \lambda\int_0^1 (f^{(m)}(t))^2 dt \tag{1.2}$$

where H is the Sobolev space $W_2^m = \{f | f^{(\nu)}$ are absolutely continuous for $\nu = 1,2,...,m-1$ and $f^{(m)} \in L_2[0,1]\}$. The smoothing parameter λ controls the tradeoff between "closeness" to the data as measured by the first term and "roughness" of the solution as measured by the second term. As is well known, the choice of the smoothing parameter is crucial to spline estimates. We will adopt the Generalized Cross Validation method which was introduced by Craven and Wahba (1979) to estimate λ since GCV method has been proven to provide a nice estimate of λ theoretically and numerically. For theoretical results on the efficiency of GCV estimate of λ, see Craven and Wahba (1979), Speckman (1982) and Li (1985,1986).

For problems with discontinuities, by choosing H to be the Sobolev space W_2^m augmented by a jump space consisting of some appropriate truncated polynomials with derivatives defined almost everywhere, the estimate g_λ of g can be shown to be

$$g_\lambda = \sum_{i=1}^{n} c_i \xi_i + \sum_{j=1}^{m} d_j \phi_j + \sum_{k=1}^{q} \theta_k \gamma_k, \tag{1.3}$$

where the c's, d's and θ's are real numbers, and

$$\xi_i(t) = \int_0^1 \frac{(t-u)_+^{m-1}(t_i-u)_+^{m-1}}{((m-1)!)^2} du, i=1,...,n, \tag{1.4}$$

$$\phi_j(t) = \frac{t^{j-1}}{(j-1)!}, \; j=1,2,....,m, \tag{1.5}$$

$$\gamma_k(t) = \frac{(t-\alpha_k)_+^{d_k}}{d_k!}, \; k=1,2,....,q, \tag{1.6}$$

with $(x)_+ = \max(x, 0)$. Note that the jump function $\gamma_k^{(d_k)}(t)$ is discontinuous at α_k. Furthermore, the break points α_k's need not be distinct. See Shiau (1985) or Shiau (1987) for details.

Let Σ be the n by n matrix with (i,j)–th entry $\xi_j(t_i)$, T_p be the n by m matrix with (i,j)–th entry $\phi_j(t_i)$ and T_d be the n by q matrix with (i,k)–th entry $\gamma_k(t_i)$. Note that the polynomial basis $\{\phi_j\}$ and the truncated polynomial basis $\{\gamma_k\}$ are all in the null space of the smoothing functional $J(f) = \int_0^1 (f^{(m)}(t))^2 dt$. Letting $T = [\, T_p \; T_d \,]$ and $\beta = (\, d_1,....,d_m, \theta_1,..... \theta_q \,)^t$, it can be shown that (1.2) is equivalent to

$$\min_{c,\beta} \frac{1}{n}\|y - (\Sigma c + T\beta)\|^2 + \lambda c^t \Sigma c, \tag{1.7}$$

which is equivalent to solving the following linear system of equations:

$$\begin{cases} (\Sigma + n\lambda I)c = y - T\beta \\ T^t c = 0. \end{cases} \tag{1.8}$$

This is the same form as for ordinary smoothing splines. It is well known that the solution is unique provided that T is of full rank. Note that g_λ is a linear estimate since it can be expressed by $g_\lambda = A(\lambda)\, y$, where

$g_\lambda = (g_\lambda(t_1), g_\lambda(t_2),..., g_\lambda(t_n))^t$ and the "hat matrix"

$$A(\lambda) = \Sigma M^{-1}(I - T(T^t M^{-1} T)^{-1} T^t M^{-1}) + T(T^t M^{-1} T)^{-1} T^t M^{-1}$$

with $M = \Sigma + n\lambda I$.

Numerous algorithms are available for ordinary smoothing splines and partial splines. Reinsch's algorithm (Reinsch, 1967) is an O(n) algorithm for computing the spline estimate g_λ if λ is fixed. The difficulty of computing the GCV function

$$V(\lambda) = \frac{\frac{1}{n} \| (I - A(\lambda))y \|^2}{(\frac{1}{n} tr(I - A(\lambda)))^2} \qquad (1.9)$$

lies in the computation of the trace in the denominator.

Wendelberger (1981) developed a numerical algorithm for obtaining the GCV estimate λ and spline estimates of functions of several variables. This algorithm is practical for moderate data set problems, but is not practical for large data set problems. The reason is that it involves an eigenvalue–eigenvector decomposition of an n–d by n–d matrix, where d is the fixed dimension of the null space of the smoothing functional. The complexity of the algorithm is $O(n^3)$ due to that costly decomposition.

Bates and Wahba (1983) suggested some methods to reduce the computing burden, including using basis functions (e.g. B–splines) of a subspace of smaller dimension or a truncated singular value decomposition to handle large data set problems. Recently, Bates et al. (1986) have developed a public domain software package called GCVPACK for computing smoothing splines and partial splines for the multivariate case. The procedures require $O(n^3)$ operations.

Utreras (1980) proposed an approximation to the trace of $A(\lambda)$ in the case of equally spaced data. This approximation requires O(n) operations for its calculation, so the GCV can be obtained cheaply. Utreras (1981) considered the case of not necessarily equally spaced data and obtained an approximation that has an initial overhead of finding the lowest n eigenvalues of a 2n by 2n band matrix of bandwidth 5 (for m = 2) which requires $O(n^2)$ operations.

Based on the special structure of cubic splines, Silverman (1984) modified Utreras' approximation and developed a linear time procedure called "Asymptotic Generalized Cross–Validation" to obtain the smoothing parameter.

Elden (1984) modified the method of computing GCV function. Instead of computing the singular value decomposition (SVD) of an n by p matrix, he used a bidiagonalization which in fact is the first part of a singular value decomposition. He then showed that starting out from the bidiagonal decomposition, the GCV function can be computed in O(n) operations. He claimed that if n and p are close, then the computation of this algorithm usually requires less than one third of the work for the full SVD. However, the bidiagonalization for an n by p matrix still needs $O(np^2)$ operations.

Recently, for computing the GCV function for the general regularization/smoothing problem, Gu et al. (1988) developed an algorithm which is based on the Householder tridiagonalization similar to Elden's (1984) bidiagonalization. This speeds up the algorithm used in GCVPACK by a factor of 6 for n large (> 500). The source code of the software package implementing this algorithm, called RKPACK, and the report on its performance can be found in Gu (1988).

Note that these fast O(n) algorithms involve some kind of approximation. They either approximate the solutions from a subspace of lower dimension, truncate some smaller eigenvalues, or approximate the trace of $A(\lambda)$. In the following sections, based on the special structure of the covariance matrix Σ, we propose efficient algorithms for the one dimensional case to compute spline estimates and $V(\lambda)$ for the functions with or without discontinuities.

Recently, Hutchinson and de Hoog (1985) developed a linear time procedure to compute the ordinary GCV smoothing spline based on Reinsch's algorithm for computing the trace of $A(\lambda)$ in the general, not necessarily equally spaced or uniformly weighted case. We note that their approach, although quite different from ours, is actually based on a very similar structure.

In Section 2, we describe the special structure of the matrix Σ which inspired the construction of algorithms. In Section 3, a linear time algorithm for smoothing splines with jumps is derived for the equally spaced knots case; also a quadratic time algorithm is mentioned for the unequally spaced knots case. A simpler algorithm for ordinary smoothing splines, i.e., without jumps, is given in Section 4.

2. SPECIAL STRUCTURE OF Σ_m

Inspired by a manuscript of Wahba (1969) where a special structure of matrices to be inverted for Tchebychev splines in their most general form is exhibited, we observe that Σ_m, the covariance matrix Σ (defined in Section 1) corresponding to m, can be transformed to a symmetric (2m–1)–band matrix. This special structure of Σ_m will be used to develop efficient algorithms in Section 3 as well as in Section 4. To describe the transformation, we first define an n–m by n matrix Δ_m which transforms $\mathbf{g}=(g(t_1), g(t_2),..., g(t_n))^t$ to an (n–m)–vector corresponding to the second divided difference of g. Here we adopt the definition and notation in deBoor (1978). Denote the m–th divided difference of a function g at points $t_i, t_{i+1},..., t_{i+m}$ by $[t_i,...,t_{i+m}]g$. Assume that t_i's are all distinct (the problem of repeated observations can be resolved by averaging repeated observations and assigning appropriate weights to data points), and let Δ_m be the (m+1)–band (n–m) by n matrix with (i,j)–th entry

$$(\Delta_m)_{ij} = \begin{cases} \prod_{\substack{k=i \\ k \neq j}}^{i+m} (t_j - t_k)^{-1} & \text{for } i \le j \le i+m, \\ 0 & \text{otherwise.} \end{cases} \qquad (2.1)$$

Then

$$\Delta_m \begin{bmatrix} g(t_1) \\ g(t_2) \\ \vdots \\ g(t_n) \end{bmatrix} = \begin{bmatrix} [t_1,...,t_{1+m}]g \\ [t_2,...,t_{2+m}]g \\ \vdots \\ [t_{n-m},..., t_n]g \end{bmatrix}. \qquad (2.2)$$

For example, letting m=2, n=5 and $t_i = i/5$, we have

$$\Delta_2 = \frac{5^2}{2}\begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

We remark here that Reinsch(1967,1971) and Shiller (1984) also used this matrix in the spline context. Then the covariance matrix Σ_m can be transformed into a (2m−1)−band matrix as stated in the following proposition.

Proposition 2.1. $\Delta_m \Sigma_m \Delta_m^t$ is a symmetric (2m−1)−band matrix.

A proof of Proposition 2.1 is in the Appendix.

We can expand Δ_m to an n by n invertible matrix $\overline{\Delta}_m$ by adding m rows on the top of Δ_m. We choose the following setup to make new rows consistent with the others in Δ_m. Let $t_0 = 0$, $t_{-1} = t_1, \ldots, t_{-m+1} = t_{m-1}$ and define

$$(\overline{\Delta}_m)_{ij} = \begin{cases} \prod_{\substack{k=i-m \\ k\neq j}}^{i} (t_j - t_k)^{-1} & \text{for } i-m \le j \le i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

Then $\overline{\Delta}_m$ is a lower triangular matrix with nonzero diagonals, hence invertible.

Proposition 2.2. $\overline{\Delta}_m \Sigma_m \overline{\Delta}_m^t$ is a symmetric *(2m−1)−band matrix.*

A proof of Proposition 2.2 is in the Appendix.

3. NUMERICAL ALGORITHMS FOR SMOOTHING SPLINES WITH JUMPS

For simplicity, the subscript of Δ_m will be suppressed if no confusion can occur. Denote $\tilde{c} = (\overline{\Delta}^t)^{-1}c$, $\Upsilon = \overline{\Delta}T$, $\tilde{y} = \overline{\Delta}y$, and $\mathfrak{M} = \overline{\Delta}(\Sigma + n\lambda I)\overline{\Delta}^t = \overline{\Delta}\, M\, \overline{\Delta}^t$. Then the system of equations (1.5) is equivalent to

$$\begin{cases} \mathfrak{M}\,\tilde{c} + \Upsilon\,\beta = \tilde{y} \\ \Upsilon^t \tilde{c} = 0, \end{cases} \tag{3.1}$$

and the solution can be expressed explicitly as

$$\begin{cases} \beta = (\Upsilon^t \mathfrak{M}^{-1}\Upsilon)^{-1}\Upsilon^t \mathfrak{M}^{-1}\tilde{y} \\ \tilde{c} = \mathfrak{M}^{-1}(I - \Upsilon(\Upsilon^t \mathfrak{M}^{-1}\Upsilon)^{-1}\Upsilon^t \mathfrak{M}^{-1})\tilde{y}. \end{cases} \tag{3.2}$$

Note that by the construction of the matrix $\overline{\Delta}$ and Proposition 2.2, $\mathfrak{M}$ is a symmetric (2m+1)−band matrix. Therefore the inverse of $\mathfrak{M}$ can be computed in $O(n^2)$ operations (e.g.Dongarra et al., 1979). Moreover, note also that $\Upsilon = \overline{\Delta}T$ is a very sparse matrix due to Lemma A.1. Therefore, β can be computed without involving many entries of $\mathfrak{M}^{-1}$ for the equally spaced knots case.

We only consider the case of m=2. In the following, we develop a procedure which computes each entry of $\mathfrak{M}^{-1}$ in constant time after a linear time overhead. This leads to the linear time algorithm for the equally spaced knots case.

For $t_i = i/n$, i=1,2,....,n, and $t_{-1} = -t_1$, $t_0 = 0$, by (2.3), $\overline{\Delta}_2$ can be expressed explicitly as

$$\frac{1}{2}n^2\begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & \cdot \\ -2 & 1 & 0 & \cdot & \cdot & \cdot \\ 1 & -2 & 1 & 0 & \cdot & \cdot \\ 0 & 1 & -2 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & 0 & 1 & -2 & 1 \end{bmatrix}, \tag{3.3}$$

and

$$\overline{\Delta}\overline{\Delta}^t = \frac{n^4}{4}\begin{bmatrix} 1 & -2 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ -2 & 5 & -4 & 1 & 0 & \cdot & \cdot & \cdot & \cdot \\ 1 & -4 & 6 & -4 & 1 & 0 & \cdot & \cdot & \cdot \\ 0 & 1 & -4 & 6 & -4 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & 1 & -4 & 6 & -4 & 1 \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 & -4 & 6 & -4 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 & -4 & 6 \end{bmatrix}. \tag{3.4}$$

It can be shown that $\mathfrak{M}$ has the same 5−band structure as in (3.4) with $\mathfrak{M}_{11} = n(2+L)/24$, $\mathfrak{M}_{12} = n(1-2L)/24$, $\mathfrak{M}_{22} = n(4+5L)/24$, $\mathfrak{M}_{ii} = n(4+6L)/24$, for $i \ge 3$, $\mathfrak{M}_{i,i+1} = n(1-4L)/24$, for $i \ge 2$ and $\mathfrak{M}_{i,i+2} = nL/24$, for all $i \ge 1$, where $L = 6n^4\lambda$. Recall that the (j,i)−th entry of $\mathfrak{M}^{-1}$ can be computed as the ratio of the cofactor of the element $\mathfrak{M}_{ij}$ and the determinant of $\mathfrak{M}$. Utilizing the band pattern of the matrix $\mathfrak{M}$, we are able to compute each entry of $\mathfrak{M}^{-1}$ efficiently by the following procedure which is described in a more general form.

Let A^k be a k by k symmetric 5−band matrix (of the same form as $\mathfrak{M}$) with $A_{11} = a$, $A_{12} = b$, $A_{i,i+2} = c$, for $1 \le i \le k-2$, $A_{22} = d$, $A_{i,i+1} = e$, for $2 \le i \le k-1$ and $A_{ii} = f$, for $3 \le i \le k$. Let B^k be the k by k matrix of the lower block of A^{k+2}, that is, the 5−band symmetric matrix with $B_{ii} = f$, for $1 \le i \le k$, $B_{i,i+1} = e$, for $1 \le i \le k-1$ and $B_{i,i+2} = c$, for $1 \le i \le k-2$. Also define $\tilde{A}^k$ and $\tilde{B}^k$ to be the k by k matrices obtained by removing the last row and the j−th column of A^{k+1} and B^{k+1} respectively. Let $B^k(j)$ be the k−1 by k−1 matrix obtained by removing the first row and the j−th column of B^k. Denote by α_k, $\tilde{\alpha}_k$, β_k, $\tilde{\beta}_k$ and $\beta_{k,j}$ the determinants of matrices A^k, $\tilde{A}^k$, B^k, $\tilde{B}^k$ and $B^k(j)$ respectively. We also need to compute γ_k, the determinant of the matrix F^k (= $B^{k+1}(k+1)$) which is the k by k matrix obtained by deleting the first row and the last column of the matrix B^{k+1}.

Procedure 1. (Computing any entry of $(A^n)^{-1}$)
Step 1 Recursively compute $\{\alpha_k, k=1,2,\ldots,n\}$ and $\{\tilde{\alpha}_k, k=1,2,\ldots,n-1\}$ as follows:

$$\alpha_k = f\alpha_{k-1} - e\tilde{\alpha}_{k-1} + ce\tilde{\alpha}_{k-2} - fc^2\alpha_{k-3} + c^4\alpha_{k-4}$$

$$\tilde{\alpha}_k = e\,\alpha_{k-1} - ec\,\alpha_{k-2} + c^2\,\tilde{\alpha}_{k-2}$$

with initial conditions

$$\alpha_0 = 1,\ \alpha_1 = a,\ \alpha_2 = ad - b^2,\ \alpha_3 = \det\begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix},$$

$$\tilde{\alpha}_0 = 0,\ \tilde{\alpha}_1 = b,\ \tilde{\alpha}_2 = ae - bc.$$

Step 2. Recursively compute $\{\beta_k,\ k=1,2,\ldots,n\}$ and $\{\tilde{\beta}_k,\ k=1,2,\ldots,n-1\}$ as follows:

$$\beta_k = f\beta_{k-1} - e\tilde{\beta}_{k-1} + ce\tilde{\beta}_{k-2} - fc^2\beta_{k-3} + c^4\beta_{k-4}$$

$$\tilde{\beta}_k = e\,\beta_{k-1} - ec\,\beta_{k-2} + c^2\,\tilde{\beta}_{k-2}$$

with initial conditions

$$\beta_{-3} = \beta_{-2} = \beta_{-1} = 0,\ \beta_0 = 1,$$

$$\tilde{\beta}_{-1} = \tilde{\beta}_0 = 0.$$

Step 3. Recursively compute $\{\gamma_k,\ k=1,2,\ldots,n-1\}$ by

$$\gamma_k = e\,\gamma_{k-1} - cf\,\gamma_{k-2} + c^2e\,\gamma_{k-3} - c^4\gamma_{k-4}$$

with initial conditions

$$\gamma_{-3} = \gamma_{-2} = \gamma_{-1} = 0,\ \gamma_0 = 1.$$

Step 4. To compute $(A^n)^{-1}_{ji}$ for $j \ge i$, we first compute the cofactor of the element A^n_{ij}, $\mathrm{Cof}\,A^n_{ij}$, as

$$\mathrm{Cof}\,A^n_{11} = \beta_{n-1} - (f-d)\,\beta_{n-2},$$

$$\mathrm{Cof}\,A^n_{1j} = (-1)^{1+j}\{\beta_{n,j} - (e-b)\,\beta_{n-1,j-1} + (f-d)c\,\beta_{n-2,j-2}\},\ \text{for } j \ge 2,$$

$$\mathrm{Cof}\,A^n_{ii} = \alpha_{i-1}\,\beta_{n-i} - c^2\alpha_{i-2}\,\beta_{n-i-1},\ \text{for } 2 \le i \le n,$$

$$\mathrm{Cof}\,A^n_{ij} = (-1)^{i+j}\{\alpha_{i-1}\,\beta_{n-i+1,j-i+1} - c\,\tilde{\alpha}_{i-1}\beta_{n-i,j-i} + c^3\,\alpha_{i-2}\beta_{n-i-1,j-i-1}\},\ \text{for } 2 \le i < j \le n,$$

where $\beta_{k,j} = \gamma_{j-1}\beta_{k-j} - c\gamma_{j-2}\tilde{\beta}_{k-j} + c^3\gamma_{j-3}\beta_{k-j-1}$, for $1 \le j \le k \le n$.

Then $(A^n)^{-1}_{ji} = (\mathrm{Cof}\,A^n_{ij})/\alpha_n$.

The complexity of this procedure is easily seen to be $O(n)$ in step 1 through step 3, and computing one entry in step 4 is $O(1)$. Therefore, if we want to get the inverse of $\mathfrak{M}$, the best we can do from this procedure is $O(n^2)$ since $\mathfrak{M}^{-1}$ is in general a full matrix even if $\mathfrak{M}$ is banded. But Υ is a very sparse matrix in this application. If $m = 2$, there are only $3 + 2q$ nonzero entries where q is the number of break points. Based on that, we shall show that $V(\lambda)$, $\mathbf{c}$, and $\boldsymbol{\beta}$ can be computed in $O(n)$ operations.

(1) The solution $\boldsymbol{\beta} = (\Upsilon^t\mathfrak{M}^{-1}\Upsilon)^{-1}\,\Upsilon^t\mathfrak{M}^{-1}\tilde{\mathbf{y}}$ takes at most $(3+2q)n$ entries of $\mathfrak{M}^{-1}$.

(2) Since $\mathfrak{M}$ is a band matrix, $\mathfrak{M}\,\tilde{\mathbf{c}} = \tilde{\mathbf{y}} - \Upsilon\,\boldsymbol{\beta}$ can be solved in linear time, e.g., see Dongarra et al. (1979). Then $\tilde{\mathbf{c}}$ can be transformed back to $\mathbf{c}$ in linear time by $\mathbf{c} = \Delta^t\tilde{\mathbf{c}}$.

(3) To show that the GCV function $V(\lambda)$ can be computed in linear time, we first note that

$$I - A(\lambda) = n\lambda\,M^{-1}(I - T(T^tM^{-1}T)^{-1}T^tM^{-1})$$

$$= n\lambda\,\bar{\Delta}^t\mathfrak{M}^{-1}(I - \Upsilon(\Upsilon^t\mathfrak{M}^{-1}\Upsilon)^{-1}\Upsilon^t\mathfrak{M}^{-1})\bar{\Delta}, \quad (3.5)$$

and that the numerator and the denominator of $V(\lambda)$ are

$$\|(I-A(\lambda))\mathbf{y}\|^2 = \|\,n\lambda\,\bar{\Delta}^t\tilde{\mathbf{c}}\,\|^2 = \|\,n\lambda\mathbf{c}\,\|^2, \quad (3.6)$$

$$\mathrm{tr}\,(I-A(\lambda)) = n\lambda\,[\,\mathrm{tr}\,(\bar{\Delta}^t\mathfrak{M}^{-1}\bar{\Delta}) - \mathrm{tr}(\bar{\Delta}^t\mathfrak{M}^{-1}\Upsilon(\Upsilon^t\mathfrak{M}^{-1}\Upsilon)^{-1}\Upsilon^t\mathfrak{M}^{-1}\bar{\Delta})]. \quad (3.7)$$

To compute the trace of $\bar{\Delta}^t\mathfrak{M}^{-1}\bar{\Delta}$ ($= \mathrm{tr}\,(\mathfrak{M}^{-1}\bar{\Delta}\Delta^t)$), we actually only need the central $2m+1$ bands of $\mathfrak{M}^{-1}$ since only the central $2m+1$ bands of $\bar{\Delta}\Delta^t$ are nonzero. Also the $(m+1)$ by n matrix $\Upsilon^t\mathfrak{M}^{-1}\bar{\Delta}$ can be computed in $O(n)$ operations again by the sparsity of Υ and $\bar{\Delta}$, which shows that the second trace term can also be obtained in linear time. Thus $V(\lambda)$ can be computed in $O(n)$ operations.

Remark. For the case of the unequally spaced data points problem, $\mathfrak{M}$ does not have the regular form of (3.4). Although we still do not need the whole $\mathfrak{M}^{-1}$ matrix, we do not have a linear time algorithm to compute the required entries. However $\mathfrak{M}^{-1}$ can be computed in $O(n^2)$ operations by the band structure of $\mathfrak{M}$. Also, $\Upsilon = \bar{\Delta}T$ is still sparse. Therefore, for unequally spaced data points problems, we have a quadratic algorithm which is still more efficient than the existing cubic time algorithm in the partial spline setup.

4. EFFICIENT ALGORITHMS FOR ORDINARY SMOOTHING SPLINES

As a byproduct of developing the linear time algorithm described in Section 3, a linear time algorithm for ordinary smoothing splines for equally spaced data is available. To describe the algorithm, we first note that $\Delta T = 0$. Since $T^t\mathbf{c} = \mathbf{0}$, we can express $\mathbf{c}$ as $\Delta^t\boldsymbol{\gamma}$, for some $(n-m)$–vector $\boldsymbol{\gamma}$. Then the system of equations (1.8) can be rewritten as $W\boldsymbol{\gamma} = \Delta\mathbf{y}$ with $W = \Delta\Sigma\Delta^t + n\lambda\,\Delta\Delta^t$, which again is a symmetric $(2m+1)$–band matrix. Thus $\boldsymbol{\gamma}$ can be solved in linear time and then the solution of (1.8) is

$$\begin{cases} \mathbf{c} = \Delta^t\,W^{-1}\,\Delta\,\mathbf{y} = \Delta^t\boldsymbol{\gamma} \\ \boldsymbol{\beta} = (T^tT)^{-1}T^t(\mathbf{y} - W\,\mathbf{c}). \end{cases}$$

Also we have

$$I - A(\lambda) = n\lambda\,\Delta^t\,W^{-1}\,\Delta$$

and

$$V(\lambda) = \frac{n\|\,\mathbf{c}\,\|^2}{(\mathrm{tr}(\Delta^tW^{-1}\Delta))^2}.$$

The pattern of W is even more regular than the $\mathfrak{M}$ matrix described in section 3. In fact, W has the same pattern as the B^n matrix described in the previous section. By replacing $\{\alpha_k\}$ by $\{\beta_k\}$ and $\{\tilde{\alpha}_k\}$ by $\{\tilde{\beta}_k\}$

in the Procedure 1, we can obtain a procedure for computing any entry of the inverse of W in constant time once a linear time overhead for recursively computing three sequences of determinants is done.

Procedure 2. (Computing any entry of $(B^n)^{-1}$)
Step 1. Recursively compute $\{\beta_k,\ k=1,2,...,n\}$, $\{\tilde{\beta}_k,\ k=1,2,...,n-1\}$ and $\{\gamma_k,\ k=1,2,...,n-1\}$ as in Procedure 1.

Step 2. To compute (j,i)–th entry of $(B^n)^{-1}$ for $j \geq i$, we first compute the cofactor of the element B^n_{ij}, Cof B^n_{ij}, as

$$\text{Cof } B^n_{ii} = \beta_{i-1}\,\beta_{n-i} - c^2\beta_{i-2}\,\beta_{n-i-1}, \text{ for } 1 \leq i \leq n,$$

$$\text{Cof } B^n_{ij} = (-1)^{i+j}\{\,\beta_{i-1}\,\beta_{n-i+1,j-i+1} - c\tilde{\beta}_{i-1}\beta_{n-i,j-i} + c^3\beta_{i-2}\beta_{n-i-1,j-i-1}\,\}, \text{ for } 1 \leq i < j \leq n,$$

where $\beta_{k,j} = \gamma_{j-1}\beta_{k-j} - c\gamma_{j-2}\tilde{\beta}_{k-j} + c^3\gamma_{j-3}\beta_{k-j-1}$, for $1 \leq j \leq k \leq n$.

Then $(B^n)^{-1}_{ji} = (\text{Cof } B^n_{ij})/\beta_n$.

Remark. In the unequally spaced knots case, W is banded but is not constant on the diagonal like B^n. However, we only need the central diagonal (2m+1) bands of W^{-1} to compute the trace of $I-A(\lambda)$. Hutchinson and de Hoog (1985) have a procedure to calculate these bands in $O(n)$ operations. Thus a linear time algorithm is also available.

APPENDIX

In this appendix we give proofs of Propositions 2.1 and 2.2.

Denote $\mathbf{v} = (v_1, v_2, \ldots, v_n)^t$, the i–th row of Δ by $\Delta^{(i)}$.

Lemma A.1. $\Delta^{(i)}\mathbf{v} = 0$, provided that $v_i, v_{i+1}, \ldots, v_{i+m}$ can be interpolated by a polynomial of degree less than or equal to m–1.

Proof: Let $v_i, v_{i+1}, \ldots, v_{i+m}$ be interpolated by a polynomial p of degree less than or equal to m–1. Thus we have $p(t_j) = v_j$, for $j = i, \ldots, i+m$. It is well known that $\Delta^{(i)}\mathbf{p} = 0$, where $\mathbf{p} = (p(t_1), p(t_2), \ldots, p(t_n))^t$. Since $\Delta_{ij} = 0$ for $j < i$ or $j > i+m$, we have $\Delta^{(i)}\mathbf{v} = 0$.

Proof of Proposition 2.1. Since the symmetry is obvious, it suffices to show the (i,j)–th entry of $\Delta_m \Sigma_m \Delta_m^t$ is 0 for all $j \geq i+m$. Note that

$$(\Delta\Sigma\Delta^t)_{ij} = \sum_{k=1}^{n}\sum_{s=1}^{n}\Delta_{ik}\Sigma_{ks}\Delta_{js} = \frac{1}{((m-1)!)^2} \times \int_0^1 \Big[\sum_{k=i}^{i+m}\Delta_{ik}(t_k-u)_+^{m-1}\Big]\Big[\sum_{s=j}^{j+m}\Delta_{js}(t_s-u)_+^{m-1}\Big]\,du.$$

If $j \geq i+m$, then since the t_s's in the second sum are no smaller than the t_k's in the first term, the integral can be rewritten as

$$\int_0^1 \Big[\sum_{k=i}^{i+m}\Delta_{ik}(t_k-u)_+^{m-1}\Big]\Big[\sum_{s=j}^{j+m}\Delta_{js}(t_s-u)^{m-1}\Big]\,du.$$

Then by Lemma A.1, the second sum in the integrand is 0.

Proof of Proposition 2.2. Let U be the m by n matrix formed by the top m rows of $\bar{\Delta}_m$. Note that $U_{ij} = 0$ for $i < j \leq n$, $i = 1, 2, \ldots, m$. Then

$$\bar{\Delta}\Sigma\bar{\Delta}^t = \begin{bmatrix} U\Sigma U^t & U\Sigma\Delta^t \\ \Delta\Sigma U^t & \Delta\Sigma\Delta^t \end{bmatrix}.$$

Since $\Delta\Sigma\Delta^t$ is a symmetric 2m–1 band matrix, it suffices to show that $(U\Sigma\Delta^t)_{ij} = 0$, for $i \leq j \leq n-m$, $i = 1,2,\ldots,m$. By the same argument as in the proof of Proposition 2.1, we have $(U\Sigma\Delta^t)_{ij}$ equals to $1/((m-1)!)^2$ times

$$\int_0^1 \Big[\sum_{k=1}^{i} U_{ik}(t_k-u)_+^{m-1}\Big]\Big[\sum_{s=j}^{j+m}\Delta_{js}(t_s-u)^{m-1}\Big]\,du,$$

again, which is 0 if $j \geq i$.

ACKNOWLEDGEMENT

The author would like to thank Professor Paul Speckman for his careful reading of this paper and very useful suggestions which greatly improve the proofs of Propositions 2.1 and 2.2 as well as the quality of the paper. This research was supported by NSF under Grant No. DMS–8709977, by ONR under Contract No. N00014–77–C–0675 and by NASA under Grant No. NAG5–316.

REFERENCES

Bates, D.M., Lindstrom, M., Wahba, G. and Yandell, B. (1987), "GCVPACK–Routines for Generalized Cross Validation," *Commu. Statist. Simul. Comput.*, 16, 263–297.

Bates, D.M. and Wahba, G. (1983), "A Truncated Singular Value Decomposition and Other Methods for Generalized Cross–Validation," Technical Report #715, Dept. of Statistics, Univ. of Wisconsin–Madison.

Craven, P. and Wahba, G. (1979), "Smoothing Noisy Data with Spline Functions: Estimating the Correct Degree of Smoothing by the Method of Generalized Cross–Validation," *Numerische Mathematik*, 31, 377–403.

DeBoor, C. (1978), *A Practical Guide to Splines*, New York: Springer–Verlag. (Applied Mathematical Sciences No. 27)

Dongarra, J., Bunch, J., Mosler, C. and Stewart, G. (1979), *Linpack Users' Guide*, Philadephia, PA: SIAM.

Elden, L. (1984), "A Note on the Computation of the Generalized Cross–Validation Function for Ill–Conditioned Least Squares Problems," *BIT*, 24, 467–472.

Gu, C (1988), RKPACK — A General Purpose Minipackage for Spline Modeling," Technical Report #832, Dept. of Statistics, Univ. of Wisconsin–Madison.

Gu, C., Bates, D.M., Chen, Z. and Wahba, G. (1988), "The Computation of GCV Functions through

Householder Tridiagonalization with Application to the Fitting of Intersection Spline Models," Technical Report #823, Dept. of Statistics, Univ. of Wisconsin–Madison.

Hutchinson, M.F. and de Hoog, F.R. (1985), "Smoothing Noisy Data with Spline Functions," *Numerische Mathematik*, 47, 99–106.

Li, K.C. (1985), "From Stein's Unbiased Risk Estimates to the Method of Generalized Cross–validation," *Annals of Statistics*, 13, 1352–1377.

Li, K.C. (1986), "Asymptotic Optimality of C_L and Generalized Cross–Validation in Ridge Regression with Application to Spline Smoothing," *Annals of Statistics*, 14, 1101–1112.

Reinsch, C. (1967), "Smoothing by Spline Functions," *Numerische Mathematik*, 10, 177–183.

Reinsch, C. (1971), "Smoothing by Spline Functions II," *Numerische Mathematik*, 16, 451–454.

Shiau, J.H. (1985), "Smoothing Spline Estimation of Functions with Discontinuities," Ph.D. Thesis, also Technical Report #768, Dept. of Statistics, Univ. of Wisconsin–Madison.

Shiau, J.H. (1987), "A Note on MSE Coverage Interval of a Partial Spline Model,", *Communications of Statistics*, A16, No. 7, 1851–1866.

Shiau, J.H., Wahba, G. and Johnson, D.R. (1986), "Partial Spline Models For the Inclusion of Tropopause and Frontal Boundary Information in Otherwise Smooth Two and Three Dimensional Objective Analysis", *Journal of Atmospheric and Oceanic Technology*, 3, 714–725.

Shiller, R.J. (1984), "Smoothness Priors and Nonlinear Regression," *Journal of the American Statistical Association*, 79, 609–615.

Silverman, B.W. (1984), "A Fast and Efficient Cross–Validation Method for Smoothing Parameter Choice in Spline Regression," *Journal of the American Statistical Association*, 79, 584–589.

Speckman, P. (1982), "Efficient Nonparametric Regression with Cross–Validation Smoothing Splines," Manuscript, Dept. of Statistics, Univ. of Missouri–Columbia.

Utreras, F. (1980), "Sur Le Choix Du Parametre D'Adjustement Dans Le Lissage Par Fonctions Spline," *Numerische Mathematik*, 34, 15–28.

Utreras, F. (1981), "Optimal Smoothing of Noisy Data Using Spline Functions," SIAM Journal of Scientific and Statistical Computing, 2, 349–360.

Wahba, G. (1969) "On the Structure of Hilbert Spaces for Stochastic process Related to Splines, with Applications," Manuscript, Dept. of Statistics, Univ. of Wisconsin–Madison.

Wendelberger, J. (1981), "The Computation of Laplacian Smoothing Splines with Examples," Technical Report #648, Dept. of Statistics, Univ. of Wisconsin–Madison.

ON THE CONSISTENCY OF A REGRESSION FUNCTION WITH LOCAL BANDWIDTH SELECTION

TING YANG, University of Cincinnati

This paper studies the kernel estimators of an unknown regression function with data-based local bandwidth (LB) selection. Under the weak conditions, we discuss the uniformly strong convergence and convergence rate of kernel estimators with a local bandwidth (or an automatic local bandwidth).

1. INTRODUCTION

Let $X_1, X_2, \ldots, X_n$ be identically independently distributed random variables with unknown density function $f(x)$. We consider the kernel estimator $f_n(x)$ of the density $f(x)$ defined by the following form (Rosenblatt-Parzen type)

$$f_n(x, h_n) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{x-X_i}{h_n}\right). \tag{1.1}$$

where $K(x)$ is a real-valued Borel measurable function on $\mathbb{R}$ and h_n (> 0) is the bandwidth which is assumed to satisfy $h_n \to 0$ and $nh_n \to \infty$ as $n \to \infty$. If $K(x)$ is chosen to be a density function, *i.e.*,

$$\int K(x)dx = 1 \quad \text{and} \quad K(x) \geq 0, \tag{1.2}$$

$f_n(x)$ itself will be a probability density function.

Assume that (X, Y) is a pair of random variables. If $E(|Y|) < \infty$, there exists a regression function given by

$$m(x) = E(Y \mid X=x) = \frac{\int y\, g(x, y)dy}{f(x)} = \frac{r(x)}{f(x)}, \tag{1.3}$$

where $f(x)$ is the marginal density of X, $r(x) = \int y g(x, y)dy$, and $g(x, y)$ is the joint density of (X, Y). Let $(X_1, Y_1), (X_2, Y_2), \ldots$ be independent random observations with the same distribution as (X, Y). The kernel estimate $M_n(x, h_n)$ of $m(x)$ defined by

$$M_n(x, h_n) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{x-X_i}{h_n}\right) Y_i / f_n(x, h_n), \tag{1.4}$$

is determined by a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ of independent observations from the population, by a kernel function $K(x)$ and a bandwidth h_n. In (1.4), h_n is a sequence of bandwidths with $h_n \downarrow 0$ and $nh_n \to \infty$ as $n \to \infty$, and $f_n(x, h_n)$ is an estimate of the marginal density $f(x)$ of X. See Watson (1964), Nadaraya (1964) for the original definition, and Härdle and Marron (1985) for recent developments.

When sampling independently, uniform consistency results such as:

$$\begin{aligned} &\sup_{x \in \mathbb{R}} |f_n(x, h_n) - f(x)| \to 0 \text{ a. s.} \quad \text{or} \\ &\sup_{x \in \mathbb{R}} |M_n(x, h_n) - m(x)| \to 0 \quad \text{a. s.} \end{aligned} \tag{1.5}$$

were obtained under certain restrictions imposed on K and f, and under the restriction

$$\frac{nh_n}{\log n} \to \infty \text{ as } n \to \infty.$$

These results can be found in papers by Deheuvels (1974), Silverman (1978), Collomb (1979), Devroye (1981), and Härdle and Marron (1985).

In practice, the choice of the bandwidth h is one of the crucial points in applying $M_n(x, h)$. The estimator (1.4) exhibits a large variance if h is chosen small, but it has large bias if a large h is used. For this situation methods of a global selection of h was studied by Härdle and Kelly (1987).

However, when data are quite nonlinear, heteroscedastic, and nonhomogeneous, using a global bandwidth h may not be efficient. This situation motivates the study of the kernel estimators with data-based locally varying bandwidth. The corresponding kernel

estimator is

$$m_n(x,h_n(x))=r_n(x,h_n(x))/f_n(x,h_n(x)) \qquad (1.6)$$

where

$$r_n(x,h_n(x))=\frac{1}{nh_n(x)}\sum_{i=1}^{n}K\left(\frac{x-X_i}{h_n(x)}\right)Y_i \qquad (1.7)$$

$$f_n(x,h_n(x))=\frac{1}{nh_n(x)}\sum_{i=1}^{n}K\left(\frac{x-X_i}{h_n(x)}\right), \qquad (1.8)$$

and $h_n(x)$ denotes that the bandwidth is a function of x. If the density $f(x)$ is known, the estimate of $m(x)$ is simplified as

$$\overline{m}_n(x,h_n(x))=r_n(x,h_n(x))/f(x). \qquad (1.9)$$

For a non-random variable X, this type of estimator was studied by Müller and Stadtmüller (1987).

In the following, the notion of optimality always refers to minimization of the mean squared error (MSE) of the estimate for local bandwidth (LB) selection, or of the integrated mean squared error (IMSE) of the estimate for global bandwidth (GB) selection.

In Section 2, we present several results about the uniformly strong consistency of the kernel estimators with LB and we also point out the rate of this convergence. Mention that the conditions we use are weaker than in Härdle and Marron (1985) and Mack and Müller (1987).

2. Uniformly Strong Consistency

In the following sections, we make several restrictions on the kernel and the joint probability density of (X, Y) :

(C.1) K is bounded, continuous, symmetric, and has finite total variation.

(C.2) Assume that $K(x)\in \mathcal{M}_{0,k}$ (defination of $\mathcal{M}_{0,k}$ is in Müller and Stadtmüller (1987)) for $k\geq 2$ and $K^2(x)$ is integrable.

(C.3) The probability density $g(x, y)$ of (X, Y) has up to $(k+1)^{th}$ partial derivatives w. r. t. x. Define

$$G_j(x)=\int y g_1^{(j)}(x, y)dy .$$

Assume $G_j(x)$, $j=0, 1, \ldots, k+1$, exists for all x, and $G_k^2(x)$ is integrable and bounded.

(C.4) Y is a. s. bounded, i. e., there is a constant C, such that

$$|Y| \leq C \quad \text{a. s.}$$

We can show that the larger k is in the condition (C.2), the higher is the convergence rate of the estimator $m_n(x, h_n(x))$ with optimal LB w. r. t. MSE(m_n). For example, Epanechnikov's (1969) kernel

$$K(u)=\frac{3}{4}(1-u^2)\,I_{[-1,1]}(u)$$

is in $\mathcal{M}_{0,2}$. The kernel

$$K(u)=\frac{9}{8}\left(1-\frac{5}{3}u^2\right)I_{[-1,1]}(u) \quad \text{is in } \mathcal{M}_{0,4}$$

In this section, we discuss the convergence of regression estimation with LB. We define that $h_n=h_n(\tau_x)=\tau_x n^{-1/(2k+1)}$, where τ_x is a function of x and $0<a\leq\tau_x\leq b<\infty$. Both a and b are constant here. First we consider the case τ_x is non-random variable. But because τ_x will be estimated from the data, secondly, we study the properties of the estimators in case that τ_x is chosen as random variable. For simplifying our local bandwidth considerations, we assume that $f(x)\geq\varepsilon>0$ on finite interval $I=[-M, M]$, $M>0$, and the value τ_x is always contained in $[a, b]$. From now on we will write $m_n(\tau_x)$, $r_n(\tau_x)$, and $f_n(\tau_x)$ in place of $m_n(x,h_n(\tau_x))$, $r_n(x,h_n(\tau_x))$, and $f_n(x,h_n(\tau_x))$, respectively, to relieve the burden of notation.

Central to our study is the error process

$$m_n(\tau_x)-m(x), \qquad \text{for } a\leq\tau_x\leq b.$$

It can be rewritten as

$$m_n(\tau_x)-m(x) = \frac{1}{f(x)}[r_n(\tau_x)-r(x)]-\frac{r(x)}{f^2(x)}[f_n(\tau_x)-f(x)]$$
$$-[m_n(\tau_x)-m(x)]\frac{[f_n(\tau_x)-f(x)]}{f(x)}. \quad (2.1)$$

In order to facilitate our main discussions in this section, we state the following result which is established in Yang (1988). Define

$$\delta(n)=n^{-1/(2k+1)}(\log \log n/n)^{1/2}. \quad (2.2)$$

LEMMA 2.1. (*Uniform strong convergency of* LB *density estimators*). *Suppose that the property in* (C.3) *is true for density* $f(x)$ *instead of for* $g(x, y)$. *Condition* (C.1) *and* (C.2) *hold in both* (i) *and* (ii):

(i) *If* τ_x *is non-random, then*

$$\sup_x|f_n(\tau_x)-f(x)| = O[\delta(n)] \quad a.s. \quad (2.3)$$

(ii) *If* $\hat{\tau}_x$ *is random, then*

$$\sup_x|f_n(\hat{\tau}_x)-f(x)| = O[\delta(n)] \quad a.s. \quad (2.4)$$

Let τ be in $[a, b]$. According to Lemma 2.1, (2.1) can be simplified as

$$m_n(\tau)-m(x)+o[m_n(\tau)-m(x)]=\frac{1}{f(x)}[r_n(\tau) - r(x)]-\frac{r(x)}{f^2(x)}[f_n(\tau)-f(x)]. \quad (2.5)$$

In the case of $f(x)$ unknown, we replace $f(x)$ by its estimator f_n in (1.9). In other words, we have to consider the problem of the estimator m_n in (1.6) which has a random denominator. From a mathematically deductive point of view, this is a difficult feature in our studying process. Applying (2.5), the problem of convergence of m_n is simplified as the problem of convergence of r_n and f_n. The problem of convergence of f_n is done in Lemma 2.1.

Now we discuss the convergence of r_n. We state the following facts which are either established by traditional techniques or in literature. (For a good reference, see Prakasa Rao (1983, *p.*33-48)). We write

$$r_n(\tau)-r(x)=\alpha_n(x)+\beta_n(x),$$

where $\alpha_n(x)=Er_n(\tau)-r(x)$ is non-random, and $\beta_n(x)=r_n(\tau)-Er_n(\tau)$ is random.

Let $F(x, y)$ be a distribution function of (X, Y), and $F_n(x, y)$ its empirical distribution function based on an i. i. d. sample (X_1,Y_1), ... , (X_n,Y_n), i. e.,

$$F_n(x, y)=\frac{1}{n}\sum_{i=1}^{n} I_{(-\infty, x]\times(-\infty, y]}(X_i, Y_i),$$

where I is an indicator function. We rewrite

$$Er_n(\tau)=\frac{1}{h_n(\tau)}\int\int y K\left(\frac{x-u}{h_n(\tau)}\right)g(x,y)dudy$$
$$=\frac{1}{h_n(\tau)}\int\int y K\left(\frac{x-u}{h_n(\tau)}\right)dF(x,y),$$

and

$$r_n(\tau)=\frac{1}{h_n(\tau)}\int\int y K\left(\frac{x-u}{h_n(\tau)}\right)dF_n(x, y).$$

According to (C.1)-(C.4) and Taylor's expansion, we easily derive that

$$\sup_x\sup_{\tau\in[a,b]}|\alpha_n(x)|\le C_1 n^{-k/(2k+1)}b^k. \quad (2.6)$$

and

$$\sup_x\sup_{\tau\in[a,b]}|\beta_n(x)|\le \frac{C_2}{h_n(a)}\sup_{x,y}|F_n(x, y)-F(x, y)|, \quad (2.7)$$

where C_1 is constant, $C_2=\mu\sup|Y|$, and μ is the total variation of K. From the result of Kiefer (1961) for F continuous,

$$P_F\{\overline{\lim_{n\to\infty}} n^{1/2}\sup_{x,y}|F_n(x,y)-F(x,y)|/(\log \log n/2)^{1/2}=1\}$$
$$=1. \quad (2.8)$$

From (2.8), we obtain that there is constant C, such that

$$\sup_{x,y} |F_n(x,y)-F(x,y)| \le C\left(\frac{\log\log n}{n}\right)^{1/2} \quad a.s. \qquad (2.9)$$

Relations (2.7) and (2.9) prove that

$$\sup_x \sup_{\tau\in[a,b]} |\beta_n(x)| \le \frac{C_2}{h_n(a)}\left(\frac{\log\log n}{n}\right)^{1/2} \quad a.s. \qquad (2.10)$$

Comparing (2.6) and (2.10), we get the following result

$$\sup_x \sup_{\tau\in[a,b]} |r_n(\tau)-r(x)| \le C\delta(n) \quad a.s. \qquad (2.11)$$

where $\delta(n)$ is in (2.2). Therefore, we have proved

THEOREM 2.1. *Assume that the conditions* (C.1)-(C.4) *hold, then*

$$\sup_x |r_n(\tau_x)-r(x)| = O(\delta(n)) \qquad a.s.$$

In the situation of $f(x)$ known, under our conditions, it is easily implied that the convergence of $m_n(\tau_x)$ is the same as of $r_n(\tau_x)$. We know that the optimal LB choice requires knowledge at the point x of unknown functions, and is thus not available in practice. (see Yang (1988)) Some people study whether one can use a pilot estimate $\hat{\tau}_x$ of τ_x^* to form a data-driven bandwidth sequence $h(\hat{\tau}_x)$ in such a way that $m_n(\hat{\tau}_x)$ is as efficient as $m_n(\tau_x^*)$. For the kernel estimation case, Krieger and Pickands (1981) and Abramson (1982) answered in the positive. Mack and Müller (1987) proved similar results for the kernel regression case. Their methods of attack involved tightness and weak convergence of some error process. In this article, we will present strong consistency results on the more generally data-driven LB estimator $m_n(\hat{\tau}_x)$ under simpler conditions. We state

COROLLARY 2.1. *Suppose that conditions* (C.1)-(C.4) *hold . Then for any* τ_1 *and* τ_2 *contained in* $[a, b]$,

$$\sup_x \sup_{\tau_1,\tau_2} |r_n(\tau_1)-r_n(\tau_2)| = O(\delta(n)) \qquad a.s.$$

Proof. According to (2.11), we have

$$\sup_x \sup_{\tau_1,\tau_2\in[a,b]} |r_n(\tau_1)-r_n(\tau_2)| \le 2C\delta(n) \qquad a.s. \quad \blacksquare$$

COROLLARY 2.2. *Assume that* $\hat{\tau}_x$ *is a random variable and* $\hat{\tau}_x \in [a, b]$ *a.s .. τ_x is a function of x and* $\tau_x \in [a, b]$. *Suppose* (C.1)-(C.4) *hold . then*

$$\sup_x |r_n(\hat{\tau}_x)-r_n(\tau_x)| = O(\delta(n)) \qquad a.s.$$

Proof. By applying the following inequality

$$|r_n(\hat{\tau}_x)-r_n(\tau_x)| \le \sup_{\tau_1,\tau_2} |r_n(\tau_1)-r_n(\tau_2)| \quad \text{for any } \tau_1 \text{ and } \tau_2$$

$\in [a, b]$ and for any x, and Corollary 2.1, this lemma is done immediately. ∎

Under the conditions of Corollary 2.2, we have the following important fact.

THEOREM 2.2.

$$\sup_x |r_n(\hat{\tau}_x)-r(x)| = O(\delta(n)) \qquad a.s.$$

Proof. For any fixed x, and $\hat{\tau}_x \in [a, b]$, we have

$$|r_n(\hat{\tau}_x)-r(x)| \le \sup_{\tau\in[a,b]} |r_n(\tau)-r(x)|.$$

Hence, we imply the following inequality

$$\sup_x |r_n(\hat{\tau}_x)-r(x)| \le \sup_x \sup_{\tau\in[a,b]} |r_n(\tau)-r(x)|.$$

According to (2.11), the proof is done. ∎

Recalling relation (2.5) and applying Lemma 2.1, Theorem 2.1 and 2.2, we complete the proof of the uniformly strong convergence of regression estimator m_n for the case $f(x)$ unknown. The results are stated as following.

THEOREM 2.3. *(Uniform strong*

consistency of LB regression estimators). Suppose that conditions (C.1)-(C.4) hold in both (i) and (ii):

(i) *If* τ_x *is non-random contained in* $[a, b]$, *then*

$$\sup_{x \in I} \left| m_n(\tau_x) - m(x) \right| = O\{\delta(n)\} \qquad a.s.$$

(ii) *If* $\hat{\tau}_x$ *is random contained in* $[a, b]$, *then*

$$\sup_{x \in I} \left| m_n(\hat{\tau}_x) - m(x) \right| = O\{\delta(n)\} \qquad a.s.$$

where I is finite interval.

REFERENCES

ABRAMSON, I. (1982). Arbitrariness of the pilot estimator in adaptive kernel methods. *J. Multi. Anal.* **12** 562-567.

COLLOMB, G. (1979). Conditions nécessaires et suffisantes de convergence uniforme d'un estimateur de la régression, estimation des dérivées de la régression. *C. R. Acad. Sc. Paris*, **288** Ser. A 161-164.

DEHEUVELS, P. (1974). Conditions nécessaires et suffisantes de convergence ponctuelle presque sûre et uniforme presque sûre des estimateurs de la densité. *C. R. Acad. Sc. Paris*, **278** Ser. A 1217-1220.

DEVROYE, L. (1981). On the almost everywhere convergence of nonparametric regression function estimates. *Ann. Statist.*, **9** 1310-1319.

EPANECHNIKOV, V. A. (1969). Nonparametric estimation of a multivariate probability density. *Theory Probab. Appl.* **14** 153-158.

HÄRDLE, W. AND KELLY, G. (1987). Nonparametric kernel regression estimation-optimal choice of bandwidth. *Statistics*, **18** 21-35.

HÄRDLE, W. AND MARRON, J. S. (1985) Optimal bandwidth selection in nonparametric regression function estimation. *Ann. Statist.* **13** 1465-1481.

KIEFER, J. (1961). On large deviations of the empiric d. f. of vector chance variables and a law of the iterated logarithm. *Pacific J. Math.*, **11** 649-660.

KRIEGER, A. M. and PICKANDS, J. (1981). Weak convergence and efficient density estimation at a point. *Ann. Statist.* **9** 1066-1078.

MACK, Y. P. and MÜLLER, H.-G. (1987). Adaptive nonparametric estimation of multivariate regression function. *J. Multi. Anal.* **23** 169-182.

MÜLLER, H.-G. AND STADTMÜLLER, U. (1987). Variable bandwidth kernel estimators of regression curves. *Ann. Statist.*, **15** 182-201.

NADARAYA, E. A. (1964) On estimating regression. *Theory Probab. Appl.*, **9** 141-142.

PRAKASA RAO, L. S. P. (1983). Nonparametric Functional Estimation. New York: Academic Press.

SILVERMAN, B. W. (1978). Weak and strong uniform consistency of the kernel estimate of a density and its derivatives. *Ann. Statist.*, **6** 177-184 (Add. **8** 1175-1176 (1980)).

WATSON, G. S. (1964). Smooth regression analysis. *Sankhya*, **A26** 359-372.

YANG, T. (1988). On the nonparametric estimation with local bandwidth selection. *Doctoral Dissertation, unpublished.*

Footnote:

AMS 1980 subject classification numbers. Primary 62G05; secondary 62H12.

Key words and phrases. Nonparametric regression estimation, kernel estimators, global bandwidth selection, local bandwidth selection, automatic local bandwidth selection.

VIII. SOFTWARE TOOLS FOR STATISTICS

Software for Bayesian Analysis : Current Status and Additional Needs - II

Prem K. Goel
The Ohio State University

Abstract

This article provides fairly comprehensive information about the existing software for Bayesian data analysis. An earlier version of this article is published in Goel(1988). Even though new software is being developed at a reasonable pace, the Bayesian software available for widespread usage is still in its infancy. Thus the goal of a general purpose Bayesian Statistical Analysis Package is a long way to go. Two avenues for quickly reaching this goal are discussed in the concluding section.

1. Introduction.

In May 1986, a workshop on Bayesian computing, to discuss various issues in an open forum, was organized at The Ohio State University. The two main issues discussed were (1) desirable computing environments for Bayesian statistical analysis, and (2) potentials for a Bayesian Analysis package. Although almost all the participants believed that wide-spread use of Bayesian methodology will not become a reality without an *interactive Bayesian statistical analysis package*, most agreed that it is too early to push for one. Diverse points of view also existed about the environment suitable for a future package.

However, it was suggested that future development of a package will become easy if new Bayesian software is compatible with an existing statistical package with excellent data handling and graphics capabilities, e.g.,'S©'. Development of a Bayesian '*Bulletin Board*' and '*Software Database*' accessible via networks for news and file transfers, was also suggested . These task have not been initiated as of now. Hopefully, such an initiative may be taken in the Fall '88.

The information compiled in this article was provided by the individuals listed within the parenthesis after the program name. We did not have access to any mailing list for the engineers involved in risk assessment and reliability, who have developed several special purpose Bayesian analysis programs which could be adapted for general reliability applications. Thus the listing of reliability programs is rather incomplete. On comparing similar listings in Press(198^), it is clear that impressive gains have been made in the development of software for implementing Bayesian paradigm, based on realistic specifications of prior information, via approximations, numerical analysis, and Monte Carlo integration techniques. On the other hand, it is also clear that only a few people have devoted their energy in developing Bayesian analysis software.

The available software is listed according to the following categories: general purpose data monitor (Section 2); Regression, Time Series & Econometric modeling (Section 3); Computation/Approximation of posterior distribution features (Section 4); Elicitation of prior information (Section 5); Reliability Analysis (Section 6) and Miscellaneous (Section 7). Our views on developing a general purpose Bayesian Analysis Package are given in Section 8.

2. General purpose data analysis

Program Name: **CADA** [Computer Assisted Data Analysis Monitor, 1983 (CADA Group)]
Function: CADA, a conversational language for Bayesian analysis, is a hierarchically structured system with several component groups.
Input: On-line raw data entry or data files to be loaded.
Output: Analysis for beta, two-parameter normal, and multinomial models based on conjugate priors; assessment of conjugate priors and utility functions; full rank Model I ANOVA and MANOVA for multifactor designs using conjugate or noninformative priors; simultaneous estimation of regression in m-groups; psychometric methods; EDA; probability distribution and actuarial functions.
Language: BASIC Compiler or interpreter required.
Machines: DEC-PDP-11(RSTS); DEC-VAX-11(VMS), PRIME, HP-3000. IBM PC version to be released soon.
Documentation: Novick, M.L.et al.(1983), *Manual for the Computer-Assisted Data Analysis (CADA) Monitor*, Iowa City, IA: CADA Group, Inc..
Availability: Available for $600 per copy from The CADA Group, Inc., 306 Mullin Ave., Iowa City, IA 52240, Tel. # (319) 351-7200

Program Name: **BAYES PAK**(Barlow)
Function: A menu driven collection of programs for teaching simple Bayesian analysis concepts. It provides plotting capability for data and for various densities as well as analysis and simulation. It is used at UC Berkeley for an engineering statistics course.
Input: Menu driven interactive environment prompts for input parameters.
Output: The program provides plotting capability for densities involved in the conjugate Bayesian analysis of Binomial and Normal data. It can also plot two densities for different parameter specifications simultaneously.

Some simulation capability using Uniform and White noise random variables is also available.
Language: BASIC
Machines: IBM PC-AT or compatibles, IBM EGA or CGA graphics card
Documentation: Barlow, R.E, *BAYES PAK, Users Manual*, Berkeley, CA: University of California
Availability: Diskette available from Prof. Richard E. Barlow, Department of I.E. & O.R., University of California, Berkeley, CA 94720

3. Normal Linear Regression, Time Series & Econometric models.

Program Name: BATS [Bayesian Analysis of Time Series, Release 1.1, June 1987(West)]
Function: This software package provides a completely menu driven collection of functions that can be used for a variety of activities in data management, analysis and graphical displays. Bayesian approach to time series modeling and forecasting is based on a wide class of dynamic linear, and non-linear, models suitable for many types of time series data arising in industrial , economic and scientific investigations. The program allows data transformations and dynamic model definition, specifying components for smooth trends, described by polynomial functions over time, regression effects of independent variables, additive or multiplicative seasonal components and error terms as well as interactive specification of prior distributions on model components..
Input: Menu Driven interactive environment prompts for input parameters. No knowledge of APL is necessary.
Output: Interactive mode for data description and summaries and displays in numerical and graphical forms; sequential model estimation; numerical and graphical displays of features of fitted model, smoothed estimates of components including trend, growth, seasonal effects and factors, regression effects and parameters, residuals, and error variances. In addition, retrospective fit of time series and step-ahead forecasts are also available. The numerical summaries and model information can be saved on disk file or printed. Interactive manipulation of graphic displays for report production is also possible.
Language: APL*PLUS/PC© Release 6.3 or later (user must have the interpreter)
Machines: IBM PC, AT&T and compatibles with a minimum of 520K RAM
Documentation: West, M., Harrison, J. and Pole, A.(1987) BATS: A User Guide, Coventry, England: University of Warwick
Availability: Available for private or academic use for a nominal charge of 30 Pounds Sterling from the Bayesian Forecasting Group, Department of Statistics, University of Warwick, Coventry CV4 7AL, England.
Remarks: Some of the theoretical developments and applications are discussed in West, Harrison & Migon(1985) and West & Harrison(1986), Harrison & West(1987).

Program Name: BRAP [Bayesian Regression Analysis Program, Ver. 2.0 (Abowd/ Zellner)]
Function: Provides a unified package for the Bayesian analyses of the normal linear multiple regression model (MRM) with multivariate normal errors under a noninformative prior, a g-prior or a natural conjugate prior distribution. Some data transformations are built-in and IMSL© could be used for others.
Input: Control cards in JCL format. Data files loaded thru JCL.
Output: Updates the prior parameters; provides standard posterior information; Plots raw data and residuals, marginal and bivariate contours of the prior and the posterior distributions of the regression coefficients, posterior distribution of the realized errors, posterior distribution of linear functions of coefficients; quantiles of posterior distribution for nonstandard models can be obtained via numerical integration and Monte-Carlo routines.
Language: FORTRAN-IV
Machine: IBM-MVS (may need some modifications for recent IBM compilers)
Documentation: Abowd, J.M., Moulton, B. R. and Zellner, A.(1985) *The Bayesian Regression Analysis Package, BRAP user's Manual ,Version 2.0*, H.G.B. Alexander Research Foundation, Graduate School of Business, University of Chicago
Availability: Package available from Prof. Arnold Zellner, University of Chicago, Graduate School of Business 1101 East 58th Street, Chicago IL 60637 at a very nominal cost.
Remarks: Other contributors to the development of BRAP include F. Finnegan, S. Grossman, C. Plosser, P. Rossi, A. Siow, J. Stafford, and W. Vandaele.

Program Name: BRAP-PC [Bayesian Regression Analysis Package for the IBM PC(de Alba/ Rocha)]
Function: This enhancement of BRAP also includes subroutines for Bayesian disaggregation and constrained forecasting.
Language: FORTRAN 77
Machines: IBM PC and PC compatibles.
Availability: Available from Prof. Enrique de Alba, Instituto Technologico Autonomo De Mexico (ITAM), Rio Hondo, No. 1, Mexico, D.F. 01000 at a nominal mailing & diskette charges.

Program Name: SEARCH [Seeks Extreme and Average Regression Coefficient Hypothesis (Leamer/Leonard)]
Function: A user-oriented package for Bayesian inference and sensitivity analysis that pools prior beliefs about the regression coefficients with evidence embodied in a given data set. Prior beliefs are assumed to be equivalent to a previous, but possibly fictitious data set. SEARCH offers a study of the sensitivity of the posterior estimates to changes in features of the prior beliefs expressed in terms of a fictitious data set.
Input: Formatted or free-format card-image files or on-line CRT input. Input files can be prepared on SAS©,

BMDP©, TSP©, and SPSS©. SEARCH requires a double precision version of IMSL© library.
Output: Diagnostic messages for debugging syntax errors are available. Program reports summary of prior and data information received and computes the approximate posterior mode for the regression coefficients when the prior beliefs are modeled as $\mathbf{R}\beta$ having a normal distribution with a prior mean **r** and a prior covariance matrix **V**. It also reports the sensitivity of the modal estimate to changes in **r** and **V** in the form of extreme bounds for any linear function of the parameters specified by the user.
Language: FORTRAN IV. The manual for Version 6 states that SEARCH is not completely in FORTRAN source code. Several of the subroutines for performing high precision arithmetic are object code modules (written in IBM 370 machine code). Bulk of the SEARCH is written in FORTRAN IV that is compiled at UCLA on the IBM G1 Compiler.
Machine: IBM 370/3033 .
Documentation: Leamer, E.E. and Leonard, H. B. (1985) *User's Manual for SEARCH- A software package for Bayesian inference and sensitivity analysis, Version 6.*
Availability: Available for $100 per copy from Prof. E. E. Leamer, Department of Economics, UCLA, 405 Hilgard Av., Los Angeles, CA 90024, (213) 825-1011, on an IBM OS standard label 9 track 1600 BPI tape containing four card-image files.
Remarks: This version, programmed by Arvin Stidick, differs from Version 5 in efficiency of computation and economy of input/output The Manual was largely rewritten by Thomas E. Wolff. A latest example of how SEARCH can be used is given in Leamer, E. E. and Leonard, H.B.(1983) Reporting the fragility of Regression Estimates, *The Review of Economics and Statistics* .

Program Name: MICRO EBA [Micro computer version of SEARCH(Fowles)]
Function: This main program is the micro computer version of the above program SEARCH
Language: GAUSS
Machine: Any personal computer running GAUSS software package Version 1.46 or higher.
Availability: Available free of charge from Prof. Richard Fowles, Department of Economics, Rutgers University, Newark, NJ 07102.

Program Name: BRP [Bayesian Regression Program (Bauwens)]
Function: This main program performs Bayesian regression analysis for various standard econometric models, discussed in Dreze(1977). The prior beliefs are modeled as Poly-t densities evaluated via the program **PTD**.
Input: Raw data as card-image files. Input is echoed as output.
Output: Posterior parameters, precision & standard deviations, and marginals of regression coefficients; classical regression analysis, posterior residuals and predictive density function of the dependent variable; conditional posterior with given precision, conditional posteriors of some regression coefficients given the others, marginalized over the precision.
Language: FORTRAN 77
Machine: IBM 370/158 at the University of Louvain. In near term, a PC version is possible.
Documentation: Bauwens, L. and Tompa, H. (1977) *Bayesian Regression Program (BRP)*, CORE User's Manual Set # A-5, and Tompa, H.(1977) *Poly-t Distributions (PTD)*, CORE User's Manual Set # C-9.
Availability: Available for 5,000 Belgium Francs from Prof. Luc Bauwens, CORE, 34 Voie Du Roman Paays, B-1348 Louvain-La-Neuve, Belgium.
Remarks: These programs have been developed by H. Tompa under the guidance of Profs. Jacques Dreze and Jean-Francois Richard and with assistance from Luc Bauwens, Jean-Paul Bulteau and Philippe Gille.

Program Name: BARMA [Fully Bayesian Analysis of ARMA Time Series Models(Monahan)]
Function: A collection of main program and subroutines carries out the Bayesian Analysis for ARMA time series models using natural conjugate priors as described in Monahan(1983).
Output: Programs compute the posterior and predictive distributions of parameters for a given set of ARMA models using the natural conjugate prior. Graphical displays are obtained via SAS/GRAPH.
Language: FORTRAN 66
Machine: Portable
Documentation: Monahan, J.(1980) '*A Structured Bayesian Approach to ARMA time series models*, I,II,III', Technical Reports, Department of Statistics, North Carolina State University, Raleigh, NC.
Availability: The package available on tape from Prof. John Monahan, Department of Statistics, North Carolina State University, P.O. Box 8203, Raleigh , NC 27695 at a nominal charge.

Program Name: Sampling the Future (Thompson)
Function: This program simulates the predictive distribution of a set of future observations via Monte Carlo methods as discussed in Thompson (1986).
Output: The main program and subroutines provide a Monte-Carlo histogram for the predictive distribution of a future observation or a scattergram of samples from the predictive distribution of a pair of future observations. The program allows as many as 10 ARMA parameters in up to 3 AR factors and up to 3 MA factors. Thus multiplicative seasonal factors and the difference factors may be used in the model. Estimation step allows either a diffuse or a conjugate normal/ gamma prior distribution.
Language: FORTRAN 77 ANSI standard.
Machine: The program runs on any machine with standard FORTRAN 77 compiler and IMSL© library. Future extensions will requires a graphics terminal. The program will run on a PC with a math co-processor. A PC-AT type machine with a hard disk is recommended.
Availability: Diskette available for $10 from Prof. Patrick Thompson, Faculty of Management Sciences, The

Ohio State University, 1775 S. College Road, Columbus OH 43210.
Remarks: Future enhancement plans include a graphic display of predictive distributions and to add the algorithm for prediction from a set of ARMA models given in Monahan (1983).

Program Name: Bayes & Empirical Bayes Shrinkage Estimation of Regression Coefficients (Nebebe)
Function: The program computes Bayes and empirical Bayes Estimates for a multiple normal linear regression model in which the prior for the regression coefficients and the precision is modeled as a hierarchical normal with mean μ and precision τ^2. The hyperparameters are assumed to have various diffuse distributions.[see Nebebe, F. and Stroud, T.W. F.(1986).]
Language: FORTRAN, requires access to NAG© library.
Documentation: No separate documentation is available. The details are given in Nebebe, F. (1984) Ph. D. thesis, Department of Mathematics and Statistics, Queen's University, Kingston, Canada.
Availability: Available from Prof. F. Nebebe, Dept. of Decision Sc. and MIS, Concordia University, 1455 De Maisonnevue Blvd. West, Montreal, Quebec H3G1M8, Canada
Remarks: This program provides no extra capability beyond BRAP, SEARCH or BAP. But it may be useful for individuals who do not.have access to IMSL package.

Program Name: SHAZAM [General Econometrics program (White)]
Function: The program provides a portable FORTRAN program for general econometric modeling. PC version for $250, main frame version for $500-900. The author promises that the next version will include a Bayesian Inequality regression.
Availability: Available from Prof. Kenneth J. White, Economics Department, University of British Columbia, Vancouver, B.C. Canada.

Program Name: BTS [Bayesian Time series (Carlin/Dempster)]
Function: This program package carries out computations for Bayesian estimation of unobserved components('seasonal'/'nonseasonal') in monthly time series under a class of Gaussian Mixed models as described in Carlin, Dempster and Jonas(1985). It uses likelihood based methods for estimation of model parameters.
Output: The program provides posterior estimates of model parameters. A non-portable version for the Apollo DN600 workstation has many graphics capabilities.
Language: FORTRAN 77 (Standard ANSI)
Documentation: Description of the program is available in Carlin, J. B.(1987) Ph.D. Thesis, Department of Statistics, Harvard University
Availability: Available free of charge from Prof. A.P. Dempster, Department of Statistics, Harvard University, Science Center, 1 Oxford Street, Cambridge, MA 02138.

Program Name: PROC SEQ [Sequential Scoring Algorithm(Blattenberger)]
Function: The function performs iterative computation of forecasting distribution for the dependent variable of a normal linear model with a normal-gamma prior distribution or optional g-priors. Scores for five different scoring rules are also computed.
Language: STAT80 Procedure; being converted to SAS© PROC MATRIX.
Availability: Available free of charge from Prof. Gail Blattenberger, Department of Economics, University of Utah, Salt Lake City, UT.

Program Name: MAXENT [Data Analysis by Maximum Entropy Principle Version 1.17 (Jaynes)]
Function: This beta version of MAXENT provides fitting of an incompletely specified linear model of the form $\mathbf{Y}=\mathbf{X}\,\mathbf{F}$, where the data vector is **Y**, the 'smearing matrix' **X** is known but not of full rank and the elements of the vector **F** are non-negative adding to 1. The Maximum Entropy Principle, see Jaynes(1983) finds the solution which maximizes the entropy of the probability distribution of **F**.
Input: This interactive program requires the input of accuracy level for constraints satisfaction.
Output: The optimal solution is obtained iteratively, with access to the output for each iteration.
Language: BASIC
Machines: IBM PC and compatibles. An ASCII source code file is also on the diskette for transporting the program to other micro computers.
Documentation: Help file and Manual on diskette.
Availability: Available free from Prof. Ed T. Jaynes, Department of Physics, Washington University, St. Louis, MO 63130.

The programs briefly discussed below have been written for specific applications of linear models.

Program Name: RECONDA (Braithwait, Steven)
Function: This C program incorporates engineering prior estimates of appliance level electricity consumption into a statistical analysis of household hourly consumption via a hierarchical linear model. The modeling details are given in Caves, Herriges,Train, Windle(1987).
Machines: IBM PC and PC compatibles
Availability: The program will be distributed free of charge by EPRI, P.O. Box 10412, Palo Alto, CA 94303 to EPRI member utilities, government and academic institutions.

Program Name: Statistical Cost Allocation (Wright, Roger)
Function: This FORTRAN 77 program implements the indirect cost allocation methodology based on a multiple linear model as described in Wright(1983).
Documentation: The program description and listing are given in Wright, R. and Oberg, K.(1983) *The 1979-80 University of Michigan Heating Plant and Utilities Cost*

Allocation Study, Working Paper #352, Graduate School of Business Administration, The University of Michigan.
Availability: Available free of charge from Prof. Roger Wright, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI 48109.

4. Computation/Approximation of Posterior Distribution Features.

Program Names: BAYES FOUR & *gr* (Smith, A.F.M.)
Function: The Bayes Four system consists of a library of subroutines, primarily intended for numerical computation of multiple integrals in interactive mode. Posterior distribution's features can be evaluated for a practical implementation of the Bayesian paradigm for up to 6 parameters using numerical integration procedures and up to 20 parameters using Monte Carlo integration. The *gr* library consists of subroutines for an interactive color graphics system which can be used to reconstruct and display output of the Bayes Four system. For reference, see Smith, Skene, Shaw, Naylor, and Dransfield(1985).
Input: Solving an inference problem requires writing main program for calling Bayes Four and *gr* subroutines.
Output: The posterior moments and marginals can be evaluated by calling these menu driven subroutines. The *gr* package can be used to provide graphical displays of the univariate and bivariate marginal posterior densities and predictive densities from outputs of Bayes Four.
Language: Bayes Four in FORTRAN 77; *gr* in 68000 assembler, C and FORTRAN77.
Machines: BAYES FOUR for SUNIII or APPOLO workstations However *gr* has not been configured for any standard graphics system or workstation yet
Documentation: Naylor, J. C. and Shaw, J. E. H.(1985) *BAYES FOUR- User Guide* ; Naylor, J. C. and Shaw, J. E. H.(1985) *BAYES FOUR- Implementation Guide* ; Shaw, J. E. H. (1985) *gr User Guide*. All these are technical reports from the Nottingham Statistics Group, Department of Mathematics; University of Nottingham .
Availability: Available from.Prof. Adrian Smith, Department of Mathematics, University of Nottingham, Nottingham, U.K.NG7 2RD(cost for academic use $200)
Remarks: (i) For application of this system to some interesting applied problems in pharmaceutical industry, see Racine, Grieve, Fluhler, and Smith (1986) . (ii) An enhanced version of BAYES 3.5 is available from Prof. L.D. Perrichi, Department of Mathematics and Computer Science, Simon Bolivar University, Apartado 80659, Caracas 1080A, Venezuela.

Program Name: Simple Importance Sampling [Computation of Posterior moments and densities via Monte Carlo Integration (van Dijk)]
Function: This program approximates multiple integrals that arise in the posterior moments and marginal densities of parameters of interest in econometric and statistical modeling, via importance sampling Monte Carlo integration.
Language: FORTRAN 77
Documentation: The algorithm, program listing and some examples are given in van Dijk, H. K., Hop, J. P. and Louter, A. S.(1986) *An algorithm for the computation of Posterior moments and densities using simple importance sampling*. Econometric Institute Report 8625/A, Erasmus University, Rotterdam.
Availability: Available from Prof. Herman K. Van Dijk, Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738- 3000 Dr., Rotterdam, The Netherlands.
Remarks: Some standard programs for the method of mixed integration [see, van Dijk, Kloek and Boender(1985)] are under preparation by Prof. van Dijk.

Program Name: Monte Carlo Integration (Geweke)
Function: A collection of programs using some interesting methods for constructing Importance Sampling density derived from the asymptotic sampling theoretic densities of the m.l.e., which are more flexible than the multivariate Student-t density used in van Dijk program. It has built in diagnostics for the convergence of the numerical approximation to the true values almost surely. Some applications of this methodology are also give on the diskette.
Language: FORTRAN 77 with a double precision version of IMSL©.
Machines: VAX-VMS or any other machine with FORTRAN compiler.
Availability: Available on diskettes from Prof. John Geweke, Institute of Statistics and Decision Sciences, Duke University, Durham, NC 27706.

Program Name: BAYES3/3D [Multiparameter Univariate Bayesian Analysis using Monte Carlo Integration (Stewart)]
Function: Bayesian inference for univariate response variable using Monte-Carlo integration. Up to nine parameters allowed. Can handle usual random sampling data, interval data, censored data, binomial data at different stresses or times.
Input: Data and control cards as card-image files.
Output: Displays posterior means and percentile curves, hazard rate functions, or probability of failure(response) versus stress (dose) or time. (References: Stewart, L. (1979, 83, 85).
Language: FORTRAN 77
Machines: A graphics terminal is highly desirable but not absolutely necessary. Need DISPLA graphics software. GKS and DI-3000 versions are being written.
Documentation: Stewart, L. (1987) *User's Manual for* BAYES3/3D, A program for multiparameter univariate Bayesian analysis using Monte Carlo integration.
Availability: The program was developed under various Federal contracts at Lockheed-Palo Alto Research Laboratory, Palo Alto CA 94304. Dr. Leland Stewart, will provide the tape in individual cases, on permission from Lockheed.

Program Name: LINDLEY.BAS (Sloan)

Function: This BASIC subroutine performs algebraic manipulation and constructs the expanded formula for use of approximating the ratio of two integrals, required in the evaluations of the posterior distribution's features, as discussed in Lindley(1980).
Input: The program prompts for the number of parameters to be estimated.
Output: The printout gives the complete algebraic equation needed to approximate the ratio of integrals.
Language: MS BASIC
Machine: IBM PC or compatibles. Special printing customized for EPSON series of printers.
Availability: Available free of charge from Prof. Jeff A. Sloan, Department of Statistics, University of Manitoba, Winipeg, Manitoba, Canada R3T 2N2.

Program Name: SBAYES (Tierney)
Function: The system consists of S©-functions to compute approximations of posterior means, variances and marginal densities that are generally more accurate than Lindley's Method mentioned above [see for reference: Tierney and Kadane(1986)].
Language: FORTRAN 77 and C. Requires access to the S© package for implementation.
Availability: Available free of charge from Prof. Luke Tierney, School of Statistics, University of Minnesota, Minneapolis, MN 55113.

5. Elicitation of Prior Information.

Program Name: BAYES (Schervish)
Function: This program elicits priors and finds posterior and predictive distributions for samples from normal or binomial data with natural conjugate priors or mixed conjugate plus point mass priors. It also handles flat priors over bounded regions for normal data.
Language: FORTRAN IV, requires access to IMSL©.
Machine: DEC-2060. Graphics are good for GIGI© terminals only.
Availability: Available on request from Prof. Mark Schervish, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213.

Program Name: [B/D] [Beliefs adjusted by Data (Goldstein/Wooff)]
Function: This program provides an interactive, interpretive subjectivist analysis of general (partially specified, exchangeable) beliefs as described in Goldstein(1987a, b, 1988).
Output: Provides summaries of as to how and why beliefs are (i) expected to change and (ii) actually change, as well as system diagnostics based on comparison of (i) and (ii).
Language: PASCAL
Availability: Available at cost of mailing and manual production from Prof. Michael Goldstein, Department of Statistics, University of Hull, Cottingham Road, Hull, U.K.

6. Reliability Analysis.

Program Name: BASS [Bayesian Analysis for Series Systems (Martz)]
Function: This program performs a Bayesian reliability analysis of series systems of independent binomial subsystems and components for either prior or test data at the component, subsystem and overall system level. It uses a beta prior for the survival probabilities.
Language: FORTRAN 77
Machines: Portable. Requires DISPLA© software package for graphics.
Availability: Free of charge from Dr. Harry F. Martz, Group S-1, MS F600, Los Alamos National Laboratory, Los Alamos, NM 87545.

Program Name: BURD [Bayesian Updating of Reliability Data (Martz)]
Function: The program performs Bayesian updating of Binomial and Poisson likelihood with a natural conjugate prior or a lognormal prior for the parameter. The updating for lognormal prior is done via Monte Carlo integration. These models are used in nuclear industry. The program is a proprietary of Babcox and Wilcox Inc.
Documentation: Ahmed,S., Metcalf, D.R., Clark, R.E. and Jacobsen J.A. (1981) BURD- *A Computer program for Bayesian updating of reliability data* , NPGD-TM-582, Babcox and Wilcox Inc., Lynchburg, VA.

Program Name: IPRA [An Interactive Procedure for Reliability Assessment, Release 2.1, (Singpurwalla)]
Function: A menu driven program performs a prior assessment based on expert opinion or informed judgement and the posterior analysis for Weibull distributed life length data in a highly interactive manner [See Singpurwalla(1988)]. It also allows the incorporation of the analyst's opinion on the expertise of the experts.
Input: On-line data entry or use of menu option to store data in a file for later use.
Output: The program computes the marginal and joint posterior densities of the Weibull parameters. The prior and posterior reliability functions for a specified time interval as well as distributions of reliability for specified mission times can be computed. These quantities can be displayed in a tabular or 2-d/3-d graphics form or saved on disk.
Language: IBM BASIC
Machines: IBM PC-XT or AT or compatibles with math co-processor and IBM enhanced or color graphics adapters.
Documentation: Aboura, K. N. and Soyer, R.(1986) *'A User's msnual for an Interactive PC-Based Procedure for Reliability Assessment.*, Tech. Report GWU/IRRA/ Serial TR-86-14, George Washington University, Washington, D.C.
Availability: The program diskette and user's manual are available from Prof. Nozer Singpurwalla, The Institute of Reliability & Risk Analysis, George Washington University, Washington, D.C. 20052 for $95.

Program Name: **IPND**[An Interactive PC-Based System for Predicting the number of defects due to fatigue in Railroad Tracks(Singpurwalla)]
Function: A menu driven program performs a Bayesian analysis of a non-homogeneous Poisson process with a Weibull intensity function in which the assessment of the prior information about the parameters is induced via an engineering model based on S-N curves, Singpurwalla(1986). The procedure is applied to prediction of the number of defects due to fatigue in railroad tracks.
Input: On-line data entry or use of menu option to store data in a file for later use.
Output: The program computes the marginal and joint posterior densities of the parameters in the Weibull intensity function. The prior and posterior distribution of the number of defects due to fatigue over a time period is also computed. These quantities can be displayed in a tabular or 2-d/3-d graphics form or saved on disk.
Language: IBM BASIC
Machines: IBM PC or compatibles with math co-processor and CGA or EGA graphics board.
Documentation: Choksy, M. and Daryanani, S.(1987) *'An interactive PC-Based System for Predicting the Number of Defects due to Fatigue in Railroad Tracks: User's manual* " Tech. Report GWU/IRRA/ Serial TR-87-3, George Washington University, Washington, D.C.
Availability: Program diskette and user's manual are available from Prof. Nozer Singpurwalla, The Institute of Reliability & Risk Analysis, George Washington University, Washington, D.C. 20052 at a nominal charge.
Remarks: This procedure and the program has been adopted by The Association of American Railroads for the analysis of fatigue defects data in railroad tracks. It is just one indication that availability of appropriate software would lead to a widespread use of Bayesian methodology.

Program Name: **PREDSIM** [Prediction and Simulation for mixtures of exponentials(Sloan)]
Function: This PL/I program performs a Monte-Carlo simulation of sampling from a mixtures of exponentials model using a method proposed by Marsaglia. It computes Bayes estimates of the systematic parameters and reliability function & predictive intervals for future observations.
Machine: Portable. Requires access to IMSL©.
Availability: Available free of charge from Prof. Jeff A. Sloan, Department of Statistics, University of Manitoba, Winipeg, Manitoba, Canada R3T 2N2.

7. Miscellaneous.

Program Name: **DISCBDIF** (Stroud)
Function: This SAS© program classifies an input record into one of the two normal populations, based on training samples from each one. It uses either Geisser's discrimination procedure or a semi-diffuse limit of conjugate priors.
Language: Requires access to SAS© package and SAS© PROC MATRIX.
Availability: Available free of charge from Prof. Thomas W.F.Stroud, Department of Mathematics and Statistics, Queen's University, Kingston, Ontario K7L3N6.

Program Name: **BPC** [Bayesian Probabilistic Classification (Bernardo)]
Function: This is a main program for implementing Bayesian linear probabilistic classification, as discussed in Bernardo(1988) . It is written to run on APPLE Macintosh. It will be supported in the future.
Language: MS FORTRAN 77
Availability: Available free of charge from Prof. Jose M. Bernardo, Department of Statistics, Faculty of Mathematics, 46071 Valencia, Spain

Program Name: **Generalized Hypergeometric Function** (Chib)
Function: This program computes the generalized hypergeometric function, which arise in the Bayes and empirical Bayes estimation of the multiple correlation coefficient with a beta prior, [see Tiwari, Jammalamadaka and Chib(1987)].
Language: Gauss
Machines: IBM PC and compatibiles with Math 8087 Co-processor and at least 512K RAM.
Availability: Available for $5 from Prof. Siddartha Chib, Department of Economics, 125 Professional Building, University of Missouri, Columbia, MO 65211.

8. Concluding Remarks.

The CADA monitor was the first and the only general purpose program for Bayesian data analysis. It has gone through several enhancements. Even though CADA was demonstrated at several SBIE seminars and is available in various machine versions, it has not been accepted as *'the package'* for Bayesian data analysis. This is mainly because all analyses in CADA are carried out under a noninformative or a simplistic conjugate prior framework. It has no numerical integration.capability, thus it precludes analysis for realistic prior specifications. Furthermore, the BASIC language does not provide today's state of the art computing environment. The graphical interfaces in CADA is almost non-existent. The package was probably installed at almost all US universities with Bayesian faculty, but has not been used extensively for teaching courses. Thus CADA has been used to a quite limited extant.

Among the participants of the Bayesian Computing Workshop at OSU, there was no interest to choose CADA as the base for the future development of a suitable Bayesian Package. The current version of CADA monitor seems to be quite obsolete to us as the basic computing environment has not changed. On the other hand, the package is now being marketed by a private

company. Depending on their future development strategy, the algorithms in CADA could become a vehicle for an acceptable system. The future plans of the CADA group should be explored before deciding on a strategy.

The implementation of the Bayesian paradigm for a realistic data analysis requires a variety of numerical integration and approximation routines. The growth of the methodology and software for this has been phenomenal. But there is a long way to go for approximation and numerical integration procedures and useful graphical displays for high dimensional problems.

The only way to develop a quickly acceptable Interactive Bayesian Software Package is to adopt some of the existing main programs and subroutines as modules in some widely used statistics package which is available for mini and micro computers and add more modules to it as the new methodologies and its software are developed. Thus one does not have to develop data management and graphics capabilities. In addition, the students and data analysts will not have to learn yet another system. It is also wise to develop all new Bayesian software so that it could be incorporated in an already existing and widely acceptable computing environment.

The strategy of writing all Bayesian software in S© compatible routines sounds appealing from the point of view of researchers in Statistics departments, where UNIX is slowly becoming a de facto operating system. This was the dominant choice of the participants in the Bayesian Computing workshop. However, S© in not accessible to a large group of statisticians and other researchers in Business schools, Economics and Engineering departments. Thus this option will limit the accessibility of the proposed system. On the other hand, it is about time that most of us agree on one option.

We believe that a suitable package for this purpose is S©-Version II, if it is supported. Otherwise, the most appropriate choice is MINITAB, since it is supported and is very widely used for teaching and data analysis. We can expect to receive some cooperation from Minitab Inc. with a suitable proposal , specially since there are tremendous prospects for additional sales. We need to quickly settle this issue if one wants to see the '**Bayesian 21st Century**'.

References

Bernardo, J.M. (1988) 'Bayesian linear probabilistic classification' in *Statistical Decision Theory and Related Topics IV* (Eds: S.S. Gupta & J.O. Berger), New York, N.Y.: Springer-Verlag

Carlin, J.B., Dempster, A. and Jonas, J.B. (1985) 'Bayesian estimation of unobserved components in time series', *Jour. of Econometrics* **30** 67-90.

Caves, D.W.,Herriges, J.A., Train, K.A. and Windle, R.J. (1987) 'A Bayesian approach to combining conditional demand and engineering models of electricity usage' *Review of Economics and Statistics*, To appear.

Geweke, J. (1987) 'Bayesian inference in econometric models using Monte Carlo integration', **DP: 87-2,** Institute of Statistics and Decision Sciences, Duke University.

Goel, P.K. (1988) 'Software for Bayesian analysis: Current status and additional needs', in *Bayesian Statistics 3* (Eds: J.M. Bernardo, M.H.DeGroot, and A.F.M.Smith), Oxford, England: Oxford University Press.

Goldstein, M. (1987a) 'Systemic analysis of limited belief specifications', *The Statistician*, **36**

Goldstein, M. (1987b) 'Can we build a subjectivist statistical package?', in *Proc. Symposium in memoriam of Bruno de Finetti*, To appear.

Goldstein, M. (1988) 'The data trajectory', in *Bayesian Statistics 3* (Eds: J.M. Bernardo, M.H.DeGroot, and A.F.M.Smith), Oxford, England: Oxford University Press.

Harrison, P.J. & West, M. (1987) 'Practical Bayesian Forecasting' *The Statistician* **36**

Jaynes, E.T. (1983) 'Prior information and ambiguity in inverse problems' *Proc. AMS-SIAM Symposium on Inverse Problems*, New York.

Lindley, D.V. (1980) 'Approximate Bayesian methods', in *Bayesian Statistics* (Eds: J.M. Bernardo, M.H.DeGroot, D.V.Lindley and A.F.M.Smith), Valencia, Spain: University Press.

Monahan, John F. (1983) 'Fully Bayesian analysis of ARMA time Series models', *Jour. of Econometrics* **31** 307-331.

Nebebe, F. and Stroud, T.W.F. (1986) 'Bayes and empirical Bayes estimation of regression coefficients', *Canadian Jour. of Statist* . **14** 267-280.

Press, S. James (1980) 'Bayesian computer programs' in *Bayesian Analysis in Econometrics and Statistics* (ed. A. Zellner), Amsterdam: North Holland.

Racine, A., Grieve, A.P., Fluhler, H., and Smith, A.F.M. (1986) 'Bayesian methods in practice: Experiences in pharmaceutical industry (with discussion), *Applied Statistics* **35** 93-150.

Singpurwalla, Nozer D. (1986) 'An interactive PC-Based system for predicting the number of defects due to fatigue in railroad tracks' GWU/IRRA/Serial **TR-86/7**, Institute of Reliability and Risk Analysis, The George Washington University.

Singpurwalla, Nozer D. (1988) 'An interactive PC-Based procedure for reliability assessment incorporating expert opinion and survival data' *Jour. American Statist. Assoc.*, **83** 43-51.

Smith, A.F.M., Skene, A.M., Shaw, J.E.H., Naylor, J.C., and Dransfield, M. (1985) 'The implementation of the Bayesian paradigm',*Comm. Statist., Theo. & Meth.* **14(5)** 1079- 1102.

Stewart, L. (1979) 'Multiparameter univariate Bayesian analysis' *Jour. American Statist. Assoc.*, **74** 684-693.

Stewart, L. (1983) ' Bayesian analysis using Monte Carlo integration and a powerful methodology for handling some difficult problems',*The Statistician*, **32** 195-200.

Stewart, L. (1985) 'Multiparameter Bayesian inference using Monte Carlo integration- Some techniques for bivariate analysis' , in *Bayesian Statistics 2* (Eds: J.M. Bernardo, M.H.DeGroot, D.V.Lindley and A.F.M. Smith), Amsterdam: North Holland

Thompson, P. and Miller, R. B. (1986) 'Sampling the future: A Bayesian approach to forecasting from univariate time series models', *Jour. of Business and Economic Statist .*, **4** 427-436.

Tierney, L. and Kadane, J. (1986) Accurate approximations for posterior moments and marginal densities', *Jour. American Statist. Assoc.*, **81** 82-86

West, M. and Harrison, P.J. (1986) 'Monitoring and adaptation in Bayesian forecasting models' *Jour. American Statist. Assoc.*, **81** 741-750.

West, M., Harrison, P.J., and Migon, H.S. (1985) 'Dynamic generalized linear models and Bayesian forecasting' (with discussion), *Jour. American Statist. Assoc.*, **80** 73-97.

van Dijk, H.K., Kloek, T., and Boender, C.G.E. (1985) 'Posterior moments computed by mixed integration', *Jour. of Econometrics* , **29** 3-18.

Acknowledgment.

This work was partially supported by the Air Force Office of Scientific Research under contract AFOSR84-0162. We would like to thank all the colleagues who kindly sent in the information about all the software listed in this report.

An outline of Arizona. *

John Alan McDonald, Dept. of Statistics, U. of Washington

July 29, 1988

1 Introduction

This paper outlines a system called Arizona, now under development at the U. of Washington. Arizona is intended to be a portable, public-domain collection of tools supporting scientific computing, quantitative graphics, and data analysis, implemented in Common Lisp[31] and CLOS (the Common Lisp Object System)[4].

Although there is substantial implementation of some of the modules described below, this paper is more a description of a design than of an actual program. One excuse for writing a paper on not-yet existing software is that Arizona is intended primarily as a research vehicle; it is hard to predict when, if ever, it will mature and stabilize to the point of robust production-quality code. However, we hope that the ideas embodied in its design are of interest in themselves and of use in future scientific computing and data analysis systems (eg. a "New New S"[2]).

Discussion of the philosophy underlying Arizona can be found in [22,23,18,21,24,32]. Briefly, the design is motivated by our belief that an ideal system for scientific computing and data analysis should have:

- One language that can be used for both for line-by-line interaction or defining compiled procedures.
- Minimal overhead in adding new compiled procedures (or other definitions).
- A language that supports a wide variety of abstractions and the definition of new kinds of abstractions.
- Programming tools (editor, debugger, browsers, metering and monitoring tools).
- Automatic memory management (dynamic space allocation and garbage collection).
- Portability over many types of workstations and operating systems.
- A community of users and developers.
- Access to traditional Fortran scientific subroutine libraries or equivalents.
- A representation of scientific data directly in the data structures of the language.

*This research was supported by the Office of Naval Research under Young Investigator award N00014-86-K-0069, the Dept. of Energy under contract FG0685-ER2500. (The system has benefited from ideas (and sometimes code) contributed by many people, including Rick Becker, Andrew Bruce, Andreas Buja, Pat Burns, John Chambers, Bill Dunlap, Robert Gentleman, Peter Huber, Catherine Hurley, John Michalak, Wayne Oldford, Jan Pedersen, Steve Peters, Werner Stuetzle, and Alan Wilks.)

- Comprehensive numerical, graphical, and statistical functionality.
- Device independent static output graphics.
- Window based interactive graphics.
- Support for efficient and concurrent access to large databases.
- Documentation and tutorials, both paper and on-line.

The first nine points (through "access to Fortran") come for free with standard Common Lisp environments. The remaining six are the research aspects of Arizona.

Because of limitations of space, for the rest of this paper we are assuming that the reader is familiar with Common Lisp and CLOS or, at least, Lisp and object-oriented programming in general. Others who wish to read this paper should review some of the references first.

1.1 The modules

Arizona is divided into a number of modules, with limited interdependencies, to permit individual modules to stabilize and be "released" before the whole system is complete.

The modules are divided into two groups: a numerical, quantitative kernel and an interactive, window-based, scientific graphics part.

The non-graphical quantitative kernel is more developed at present, because it can be implemented in an efficient, portable way using existing standards for Common Lisp and CLOS. The quantitative kernel consists of:

- *Basic Math*, which requires Common Lisp,
- *Collections*, which requires Common Lisp and CLOS,
- *Linear Algebra*, which requires Basic Math and Collections,
- *Probability*, which requires Linear Algebra,
- *Database*, which requires Collections, and
- *Statistics*, which requires Database and Probability.

The current design for the graphics part is fairly tentative. Implementation of a portable scientific graphics toolkit requires a standardized interface between Common Lisp/CLOS and the large variety of proprietary or proposed standard window systems for workstations and personal computers (eg. Symbolics Genera [36], NeWS[33], X[29], etc.). This standard (sometimes called Common Windows) is the subject of intense activity in the Common Lisp community[13, 28]. I have identified three modules:

- *Constraints*, which requires Common Lisp and CLOS. (This module might very well be part of the non-graphical kernel, but most of the applications we have in mind at present are in graphics.)
- *Quantitative Graphics*, which requires Common Windows, Collections, Constraints, and Linear Algebra.
- *Data Analysis Graphics*, which requires Quantitative Graphics and Statistics

2 The quantitative kernel

2.1 Basic Math

Basic Math consists of things that can be reasonably implemented with Common Lisp

functions and primitive Common Lisp data structures; it does not use CLOS. Included in Basic Math are: machine constants, special functions (eg. beta, gamma) extended vector operations (analogous to the BLAS[15] used in Linpack[8]), evaluation and interpolation (eg. generic continued fractions) 1d numerical integration, and basic random number generators.

2.2 Collections

The Collections module has two parts: *Abstract Sets* and *Enumerated Collections.*

Instances of an Abstract Set class are used to represent one of the sets or spaces that arise in mathematical computing. Examples are `Integer-Interval`, `Float-Interval`, and `Vector-Space`, which are used in the Probability and Linear-Algebra modules.

The Enumerated Collection classes are in part modeled on the Collection classes in Smalltalk-80[10]; instances are used for traditional compound data structures, eg. Trees, Queues, Enumerated Sets, Dictionaries, Indexes, etc. Enumerated Collections are heavily used by the Database module.

An Enumerated Collection basically serves as a framework for iterating over its elements. A simple collection might be represented by a list; more complex collections permit more efficiency for specialized access. (Eg. a time series might use a doubly linked list to give efficient access to lagged observations; discrete data might use an n-dimensional array for quick access to the cells of a contingency table.)

2.3 Linear Algebra

The Linear Algebra module is discussed in detail in [21], where it is referred to as *Cactus.* It provides approximately the same functionality as Linpack[8] and Eispack[30]. However, CLOS allows Cactus to operate at a level of abstraction chosen to match the initial, high-level, geometric descriptions of algorithms given in standard numerical analysis texts[11]. The use of object-oriented programming makes the implementation of standard algorithms (eg. a QR decomposition) easier to understand and modify than the versions in the best Fortran libraries—without sacrificing efficiency in either space or time. In addition, it is much easier to use information about regular structure, patterns of sparsity, etc., to get improved performance in special problems. Also, the higher level of abstraction permits extensions to, for example, computations on Hilbert spaces[14].

The Linear Algebra module provides: class definitions for `Vector-Spaces`, class definitions for `Vector-Transformations` (`Matrix`, `Positive-Definite-Matrix`, `Householder`, `Product`, etc.), methods for the protocol corresponding to the algebra of linear transformations (`transform`, `compose`, `scale`, `add`), methods for "matrix" decompositions (LU, QR, LQ, SVD, eigen, etc.), and the ability to solve systems of linear equations and least squares problems using a generic `pseudo-inverse` function that can be applied to any linear transformation.

2.4 Probability

Inference and Monte Carlo simulation (including Bootstrapping) are supported in a unified framework through a protocol for `Probability-Measure` classes. Probability measure objects are responsible for generating samples from themselves, computing their quantiles, and computing the probabilities of appropriate sets, including tail

probabilities. The defined probability measure classes includes the standard one- and higher-dimensional parametric densities and discrete distributions, and non-parametric measures, either resulting from density estimates or the empirical measure of a data set. (It's worth noting that simple descriptive statistics like mean, median, etc., are generic functions in the probability measure protocol and are applied to data sets by viewing them as empirical distributions.)

2.5 Database

The Database module has two parts. The first concerns the representation of statistical data by collections of objects and is fairly well developed. The second is concerned with providing true database facilities: efficient concurrent access to large (gigabyte) collections of objects whose identities persist beyond the lifetime of a particular Lisp address space. The second part is a major research topic in the database and object-oriented programming communities[25,26].

2.5.1 Collections of objects

In most statistical packages, data sets are represented as 2 dimensional arrays of floating point numbers. Each row represents an individual and each column represents a variable. This is an awkward representation, for example, for categorical data, and for data sets with more complicated structure, such as clustering trees. It is impossible to represent simple, but important, contextual information, such as the fact the a negative value for height must be an error or that height at age 2 should be greater than height at age 1. An array representation makes it difficult to sort and select subsets without losing track of important correspondences, such as the fact the row 17 in the array of subsurface coal producers represents the same company as row 25 in the array of all coal producers and average sulfur content is column 3 in subsurface coal producers and column 5 in all coal producers.

In Arizona, statistical data is represented by collections of objects. The advantages of this are discussed in detail in [18]. Individuals are represented by objects, instances of CLOS classes. Variables are represented by generic functions. A dataset is represented by a collection, typically a list or one-dimensional array.

For example, in analyzing energy consumption data for cities in the US, the data on each city would be collected into an instance of the `City` class. A particular instance might look like:
{`City Seattle :population 450000 :cooling-degree-days 300` ⋯}.

Statistical variables are represented by generic functions. To get at the values in the slots we use automatically defined accessor functions: (`population {City Seattle}`). The use of generic accessor functions gives a unified way to refer to slots or arbitrary functions of slots; we can ask for (`log-population {City Seattle}`), where `log-population` is the obvious Lisp function.

This might seem inefficient, compared to conventional systems, where defining a new variable means adding a column to an array, because it looks as if we would have to call a procedure every time we wanted a value of the `log-population` variable. However, standard Lisp programming techniques (lazy evaluation and memo-ization [1]) make it possible to represent variables by functions, hide the additional complexity from the user, and so that the `log-population`

procedure is not called any more often than is absolutely necessary.

Each object has an identity and existence independent of any collection. So the same object can be in many collections; the unique object {`City Seattle`} would be a member of both `All-Cities` and `Northwest-Cities`. Similarly, generic functions are defined independently of any collection and can be applied to any object (for which there is a method). The independent identities maintain the important correspondences that can be hard to keep track of in an array based system.

Also, a collection may contain objects of more than one type. For example, in energy production data, it might prove useful to analyze coal and oil producers together, but to define separate coal and oil producer classes—to allow for the fact that `acres-strip-mined` is not a relevant slot for oil producers. In that case, `all-energy-producers` would contain instances of at least two different classes.

2.5.2 Persistent Objects

A true database requires objects that persist beyond the lifetime of the address space in which they were created. Arizona will be used for research into a hierarchy of functionality relating to persistent objects:

1. Making a copy of the current state of an object, in the same address space. (There are some non-intuitive difficulties in this seemingly trivial task; see [27].)
2. Saving objects to disk.
3. Automatic checkpointing
4. Objects that can undo certain changes.
5. Objects that can recover some number of previous states.
6. Objects that can recover any previous state.
7. Object identities that persist beyond a particular address space (rebooting).
8. Objects that can recover a valid state after catastrophic hardware or software failure[35].
9. Sharing objects by more than one user/address space.
10. Efficient, concurrent access to large, persistent, shared databases.

2.6 Statistics

The Statistics module represents the usual descriptive statistics by generic functions that are thought of as functionals on measures. (All the usual descriptive statistics can be thought of as functionals on measures if we consider a dataset to be a measure with total mass N.)

Simple statistical functionals take a collection and one or more variables (Lisp functions) as arguments. For example: `(median All-Cities #'log-population)`, where `All-Cities` is a `Collection` of City objects and also an `Empirical-Measure`.

`Median` returns a number; more complex statistical functional return instances of a `Description` class. A `Description` object remembers its training sample and can update itself in response to changes in the training sample. Of particular interest are `Model` objects, which are `Description`'s that are also functions.

For example, least squares linear regression takes as arguments a collection intended as the training sample, a generic

function representing the response, and a list of generic functions representing the predictors. The result of the regression is a `Regression-Model` object. The `Regression-Model` object fits itself to the training sample by 1) extracting a linear transformation by applying the predictor functions to each object in the training sample, 2) extracting a response vector by applying the response function, 3) computing a generalized inverse of the transformation via QR or SVD, and 4) applying the generalized inverse to the response vector. The regression model is also a function in the sense that it can be applied to any appropriate object (whether or not in the training sample) to predict a value for the response. In addition, the regression object is able to compute and report appropriate diagnostics and update its fit in reaction to inserting or deleting objects in the training sample or functions in the predictor list.

3 Scientific Graphics

The kernel described in the previous section is useless as a data analysis system—because it lacks any graphics. An important reason for the popularity of systems like S is their convenience and flexibility in showing pictures of data.

Our primary goal is to make it easy to improvise new kinds of plots without losing the performance needed for interactive and motion graphics. The Quantitative Graphics module supports this goal in two major ways: a defining a protocol for the representation of plots by hierarchical display objects and implementing mechanisms for maintaining constraints between the components of a display object (layout constraints) and between a window and the object(s) being shown in the window (viewing constraints).

3.1 Hierarchical Display Objects

We represent a plot as a tree of *Display-Node* objects. Every Display-Node has:

- a *parent* Display-Node. The root of the display has no parent.
- a list of *children* Display-Nodes, which is empty for terminal nodes.
- a *local coordinate system*, chosen to be convenient for describing the appearance or position of the node. For example, the local coordinate system might consist of `xyz` position coordinates, `rgb` color coordinates, a `size` coordinate, a `theta` orientation coordinate, and so on. The coordinate system is represented by an instance of an abstract set class, something like the vector spaces used in the Linear Algebra module.
- *appearance and position parameters* that allow the node to be treated as an element of the local coordinate system.
- a *local viewing transformation*, which takes local coordinates to the local coordinates of the parent. For the root node, it take local coordinates to screen coordinates—that is, pixels and pixel-values representing color. The relationship between the local viewing transformation and the coordinate systems is like the relationship of the linear transformations and vector spaces in the Linear Algebra module.
- a list of *layout constraints* which make assertions about relations between the sizes, shapes, viewing transformations, or local coordinate systems of descendants of the current node.

For efficiency in motion graphics, Display-Nodes may add:

- a *total viewing transformation*, which is obtained by composing all the local transformations between the node and the root of the tree.
- a *factoring* of the total viewing transformation into time-varying and constant parts.
- a *cache* holding the result of applying the constant factor.

For example, many implementations of rotating scatterplots implicitly factor the viewing transformation into constant translation and scaling and a time-varying rotation. If the scaling is chosen carefully, the rotation can be computed in integer arithmetic and produce exact screen coordinates, increasing the speed of rotation by as much as 100 times—at the cost of non-modular, machine specific drawing routines. However, we can implement the basic idea in a modular way, by providing methods for factoring viewing transformations analogous to the matrix decompositions provided by the Linear Algebra module. The same paradigm has been used in higher dimensional graphics[12] and is also applicable when color or shape is changing over time (rather than just position).

For efficient handling of input (deciding which node the mouse is pointing at) Display-Nodes may pre-compute and cache screen coordinates—sometimes of a single pixel, but more frequently of one or more rectangular regions.

Some Display-Nodes are *Presentations*, which means that they serve as a visible representation of some other object in the programming environment--the *subject* of the presentation. (This discussion is very loosely related to the concept of presentation given in [9] and used in the Symbolics Genera system[36] and on the Model-View-Controller user interface architecture used in Smalltalk[7].) For example, a point in a scatterplot is a presentation of a record in a data set.

A presentation is related its subject by a *viewing constraint*, discussed in the next section.

3.2 Constraints

Constraints are abstractions that arise naturally in many statistical, scientific, or graphics problems[1,17,16]. A constraint language allows the programmer to make assertions whose truth is automatically maintained in the course of subsequent computation. Spreadsheets are a widely used, if limited, form of constraint language. A full-fledged constraint language is a major research undertaking in itself [6,34,16]. We intend to implement at least two less ambitious constraints systems:

3.2.1 The Viewing Constraint

The basic idea is that a window is a view of one or more objects and should always show the current state of those objects. We have a fairly good understanding of how to implement this type of constraint. The basic technique is similar to Active Values in LOOPS[5]. The system automatically triggers appropriate computation whenever some presentation's subject is modified. The triggered computation may take place immediately or may be put off until a valid state of the presentation is needed (eg. until the window is exposed).

The viewing constraint between a presentation and its subject determines (1) how

the state of the subject is reflected in the presentation and (2) how input received by the node affects the subject. For example, if the subject is a city object in an energy consumption database and the presentation is a point in a scatterplot, the viewing constraint is responsible for supplying the presentation with, for example, population as the x coordinate, altitude as the y coordinate and average particulate ppm as a color variable. When the user selects the point with the mouse, the viewing constraint is responsible for performing the appropriate action on the subject, such as producing an editor window that lets the user inspect and possibly alter the slots and values for that particular city.

In a simple case, the presentation and subject share a *display style* object. The display style has parameters like color, size, orientation, etc. The presentation takes its appearance directly from the display style. Whenever the subject changes its display style, the presentation is automatically notified to redraw itself.

Support for the viewing constraint makes it easy to implement and generalize brushing scatterplots[20,19,3,32]. Earlier versions of brushing were based on a special plot that contained several scatterplots, each showing different variables. The basic design could not be easily extended for use in a window system where arbitrary scatterplots might be visible at any time, or to other kinds of plots besides simple scatterplots.

In Arizona, brushing is implemented in the following way: as the cursor (or brush) moves over a point in a scatterplot, the presentation is "painted" with the display style that was loaded on the brush. The constraint system causes the display style of the subject (a record in the database) to be updated automatically which in turn causes the display styles of all other presentations of that subject to be updated. A consequence of this design is that all exposed plots are automatically involved in painting. No plot needs to know what other plots are on the screen.

Extensions to other types of plots are reasonably straightforward.

3.2.2 Layout constraints

Plot layout is a more open-ended and difficult constraint problem. The idea is to provide the data analyst with a language for making and enforcing assertions about the relative sizes, shapes, or positions of the components of a plot.

A typical example—conceptually trivial but difficult to program—is centering labels around the sides of a scatterplot. The source of programming difficulty is conflicting coordinate systems. The center of the data region is naturally expressed in data coordinates. Heights and widths of label strings can usually only be determined in pixels, for a given font. The mapping of the data region into pixels cannot be determined until we know how much room is left by the labels, but we can't position the labels, choose a font, and determine the label widths and heights until we know where the data region is in device coordinates.

What we will need to support layout constraints is:

- a specification language.
- internal representation.
- general purpose satisfier.
- hooks for user supplied satisfier code.
- fast specialized satisfiers that respond to common perturbations from a solution.

- effective ways of identifying and reporting under/over constrained problems.

References

[1] H. Abelson, G. Sussman, and J. Sussman. *Structure and Interpretation of Computer Programs.* MIT Press, Cambridge, Mass., 1985.

[2] R.A. Becker, J.M. Chambers, and A.R. Wilks. *The New S Language.* Wadsworth and Brooks/Cole, Pacific Grove, CA, 1988.

[3] R.A. Becker and W.S. Cleveland. *Painting a Scatterplot Matrix: High-Interaction Graphical Methods for Analyzing Multidimensional Data.* Technical Report, AT&T Bell Laboratories., 1984.

[4] D.G. Bobrow, L.G. DeMichiel, R.P. Gabriel, S. Keene, G. Kiczales, and D.A. Moon. *Common Lisp Object System Specification.* ANSI, 1987. under preparation.

[5] D.G. Bobrow and M. Stefik. *The LOOPS Manual.* Xerox PARC, 3333 Coyote Hill Road, Palo Alto, Ca. 94304, 1983.

[6] A.H. Borning. The programming language aspects of thinglab. *ACM TOPLAS*, 3(4):353–387, 1981.

[7] B.J. Cox. *Object-Oriented Programming: An Evolutionary Approach.* Addison-Wesley, Reading, Mass., 1986.

[8] J.J. Dongarra, C.B. Moler, J.R. Bunch, and G.W. Stewart. *LINPACK Users' Guide.* SIAM, Philiadelphia, 1979.

[9] Ciccarelli E.C. *Presentation Based User Interfaces.* Technical Report 794, MIT AI Lab, 1984.

[10] A. Goldberg and D. Robson. *Smalltalk-80, The Language and Its Implementation.* Addison-Wesley, Reading, Mass., 1983b.

[11] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1983.

[12] C. Hurley. *The data viewer: a program for graphical data analysis.* PhD thesis, Dept. of Statistics, U. of Washington, 1987.

[13] Intellicorp. *Intellicorp Common Windows Manual.* 1975 El Camino Real West, Mountain View, Ca 94040-2216, 1986.

[14] G.E. Kopec. The signal representation language SRL. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-33(4):921–932, 1985.

[15] C. Lawson, R. Hanson, D. Kincaid, and F. Krogh. Basic linear algebra subprograms for Fortran usage. *ACM TOMS*, 5(3):308–371, 1979.

[16] Wm Leler. *Constraint Programming Languages.* Addison-Wesley, Reading, MA, 1988.

[17] D. Levitt. Machine tongues X: constraint languages. *Computer Music Journal*, 8:9–21, 1984.

[18] J.A. McDonald. Antelope: data analysis with object-oriented programming and constraints. In *Proc. of the 1986 Joint Statistical Meetings, Stat. Comp. Sect.*, 1986.

[19] J.A. McDonald. Exploring data with the Orion I workstation. 1982. A 25 minute, 16mm sound film, which demonstrates programs described in McDonald (1982) It is available for loan from: Jerome H. Friedman, Computation Research Group, Bin # 88, SLAC, P.O. Box 4349, Stanford, California 94305.

[20] J.A. McDonald. *Interactive Graphics for Data Analysis.* Technical Report Orion 11, Dept. of Statistics Stanford, 1982. Ph.D. thesis, Dept. of Statistics, Stanford.

[21] J.A. McDonald. *Object-oriented design in numerical linear algebra.* Technical Report 109, Dept. of Statistics, U. of Washington, 1987.

[22] J.A. McDonald and J.O. Pedersen. Computing environments for data analysis I: introduction. *SISSC*, 6(4):1004–1012, 1985.

[23] J.A. McDonald and J.O. Pedersen. Computing environments for data analysis II: hardware. *SISSC*, 6(4):1013–1021, 1985.

[24] J.A. McDonald and J.O. Pedersen. Computing environments for data analysis III: programming environments. *SISSC*, 9(2):380–400, 1988.

[25] N. Meyrowitz, editor. *OOPSLA '86 Proc.*, ACM, November 1986. SIGPLAN Notices 21 (11).

[26] N. Meyrowitz, editor. *OOPSLA '87 Proc.*, ACM, December 1987. SIGPLAN Notices 22 (12).

[27] S. Mittal, D.G. Bobrow, and K.M. Kahn. Virtual copies: at the boundary between classes and instances. In *OOPSLA '86 Proc.*, 1986.

[28] R.B. Rao. *Towards interoperability and extensibility in window environments via object-oriented programming.* Master's thesis, MIT EECS, 1987.

[29] R.W. Scheifler and J. Gettys. The X window system. *ACM TOG*, 5(2):79–109, 1986.

[30] B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe, V.C. Klema, and C.B. Moler. *Matrix Eigensystem Routines—EISPACK Guide, 2nd Edition.* Springer-Verlag, Berlin, 1976.

[31] G.L. Steele. *Common Lisp, The Language.* Digital Press, 1984.

[32] W. Stuetzle. Plot windows. *JASA*, 82(398):466–475, 1987.

[33] Sun Microsystems, Inc. *NeWS Manual.* Sun Microsystems, Inc., 2550 Garcia Ave, Mountain View, Ca. 94043, 1987. Part No. 800-1632-10.

[34] G.J. Sussman and Jr. Steele, G.L. Constraints—a language for expressing almost-hierarchical descriptions. *Artificial Intelligence*, 14:1–39, 1980.

[35] S. Thatte. Persistant memory: merging AI knowledge and databases. *TI Engineering J.*, 3(1), Jan-Feb 1986.

[36] J. Walker, D.A. Moon, D.L. Weinreb, and M. Macmahon. The Symbolics Genera programming environment. *IEEE Software*, 4(6):36–45, 1987.

AN ILLUSTRATION OF USING MACSYMA FOR OPTIMAL EXPERIMENTAL DESIGN

Kathryn Chaloner, University of Minnesota

ABSTRACT

MACSYMA is a symbolic manipulation program which can solve many algebraic problems. The feature of MACSYMA that is especially useful in optimal design problems is its ability to manipulate matrices with symbolic entries. This paper will illustrate, by an example, how it can be used in the particular problem of designing a logistic regression experiment.

1. INTRODUCTION

This paper will not attempt to review all that MACSYMA can do. The purpose of this paper is to give some indication of its usefulness by showing its use explicitly in a particular problem. The illustration uses simple commands and shows how a naive user of MACSYMA can utilize some of its very basic capabilities. The algebra in the recent papers Chaloner (1987a, b) and Chaloner and Larntz (1988) was obtained using MACSYMA and it is some of these applications which will be described.

MACSYMA is documented comprehensively in the MACSYMA reference manual. More effective introductions are given in a book by Rand (1984) and an excellent collection of worked examples by Drinkard and Sulinski (1981). Statistical applications are described by Gong (1983), Steele (1985, 1988), Rand (1988) and Chaloner (1988). In Chaloner (1988) the use of MACSYMA is illustrated in the problem of design for estimating the point at which a quadratic regression is a maximum or a minimum. The optimal Bayesian design is derived and examined, using MACSYMA. In this paper MACSYMA will be used in another, similar, problem of design for logistic regression.

MACSYMA can do many things. The aspects of MACSYMA described in this paper are some of its simplest capabilities.

2. OPTIMAL DESIGN FOR LOGISTIC REGRESSION

A logistic regression model corresponds to a binomial sampling distribution for data y. Specifically, for n_i observations taken at a value x_i of an explanatory variable, the response y_i is binomial with n_i trials and probability of success $p(x_i,\theta)$, where $\theta = (\beta_0,\beta_1)^T$ and the probability $p(x,\theta)$ is related to x by

$$p(x_i,\theta) = [1 + \exp(-\beta_0 - \beta_1 x_i)]^{-1} .$$

We think of a design as a probability measure on a compact design space $\mathcal{X}$ which puts a proportion $\eta(x_i)$ of the observations at x_i. If there are a total of n observations with n_i observations at x_i, with $\Sigma n_i = n$, then the proportion $\eta(x_i)$ is n_i/n.

The Fisher information matrix is the matrix of minus the expected value of the second derivative of the log likelihood, with the expectation taken over the sampling distribution of the data. For a design η we denote this information matrix as $nI(\theta,\eta)$, so that $I(\theta,\eta)$ is a normalized information matrix.

We can think of the design problem as choosing a measure η which optimizes some function of the information matrix. In Chaloner and Larntz (1988) designs are found which maximize the expectation, over a prior distribution on θ, of a function of the $I(\theta,\eta)$. In particular the following two criteria are maximized:

$$\varphi_1(\eta) = E \log \det I(\theta,\eta) \qquad (1)$$

and

$$\varphi_2(\eta) = - E \operatorname{tr} A(\theta)\, I(\theta,\eta)^{-1}. \qquad (2)$$

The criterion (1) is to maximize the expected value of the log of the determinant of the information matrix and the criterion (2) is to minimize (by maximizing its negative) the expected value of the weighted trace of the information matrix, where the weights may depend on θ. These critera can be justified as approximate Bayesian criteria. For the criterion of maximizing (2) the choice of $A(\theta)$ depends on what is to be estimated or predicted. Several choices of $A(\theta)$ will be discussed in Section 4.

Chaloner and Larntz (1988) show how to find optimal, or close to optimal, designs. The criterion must be evaluated using numerical integration and optimized using numerical optimization.

The numerical optimization appears to be dealt with best by fixing the number of design points, k, and finding the best design for that number of design points. The values of x_i and η_i are found numerically. A search over several values of the number of design points, k, can then be done. As k is increased the maximized criterion should become larger until, if there is an optimal design on a finite number of design points, it stays constant.

If a design is found, by numerically optimizing the criterion and searching over a number of design points, it is possible to verify that the design found corresponds to a global optimum of the criterion over all possible design measures η. A necessary and sufficient condition for a design η to be optimal is that the Fréchet directional derivative of the criterion

function, in the direction of all one point designs, is non-positive. These derivatives will be defined in Section 4 where it is demonstrated how MACSYMA can be used to find the criteria and their derivatives.

3. THE INFORMATION MATRIX

For a design measure η on k points, $x_1,\ldots,x_k$, define the function $w(x,\theta)$ as $p(\theta,x)\{1-p(x,\theta)\}$. Further define the following for i=1,...,k:

$$w_i = w(\theta,x_i)$$

$$\eta_i = \eta(x_i)$$

$$t = \sum_{i=1}^{k} \eta_i w_i$$

$$xbar = t^{-1} \sum_{i=1}^{k} \eta_i w_i x_i$$

$$s = \sum_{i=1}^{k} \eta_i w_i (x_i - xbar)^2 .$$

Note that w_i, t, xbar and s all depend on θ but this dependence has been dropped to simplify the notation.

We further define μ to be the ratio β_0/β_1 and reparameterize the problem in terms of μ and $\beta=\beta_1$. The parameter μ is the value of x at which $p(x,\theta) = 1/2$. Redefine $\theta^T=(\mu,\beta)$ then with this notation and parametrization the matrix $I(\theta,\eta)$ is:

$$I(\theta,\eta) = \begin{pmatrix} \beta^2 t & -\beta t(xbar - \mu) \\ -\beta t(xbar - \mu) & s + t(xbar - \mu)^2 \end{pmatrix} \quad (3)$$

We will use MACSYMA to show that the inverse of this matrix, $I(\theta,\eta)^{-1}$, can be expressed as:

$$\begin{pmatrix} \{s + t(xbar-\mu)^2\}/(st\beta^2) & (xbar-\mu)/(\beta s) \\ (xbar-\mu)/(\beta s) & 1/s \end{pmatrix}.$$

Figure 1 is a record of a MACSYMA session to show that this is indeed $I(\theta,\eta)^{-1}$. In the UNIX system that I use to run MACSYMA it is run by typing "macsyma" and this is shown in the first line of Figure 1 at the % prompt. Instructions typed in are labelled as (c1), (c2),.... and end with a semi-colon; corresponding output is labelled as (d1), (d2),.... . A name followed by a colon at the beginning of a command assigns the name to the resulting expression. For example, the matrix $I(\theta,\eta)$ is denoted as i in (c1) and its inverse as iinv in (c2). As MACSYMA does not recognize Greek letters, the symbols b and m are used to denote β and μ respectively.

In the (c3) command the matrices i and iinv are multiplied together to verify that they are inverses (the symbol "." denotes matrix multiplication and "*" denotes scalar multiplication). Without the expand command the resulting matrix would not be so easily identified as the identity matrix. The save command was used in (c4) to save the expressions i and iinv in file1.

4. DIRECTIONAL DERIVATIVES

The directional derivatives, as derived in Chaloner and Larntz (1988), are as follows. The derivatives for $\varphi_1(\eta)$ and $\varphi_2(\eta)$, in the direction of the design which is point mass at x, are denoted by $d_1(\eta,x)$ and $d_2(\eta,x)$ respectively. Recall that $w(x,\theta)$ is $p(\theta,x)\{1-p(x,\theta)\}$ and define $\underline{v}^T$ as $(-\beta,x-\mu)$, then:

$$d_1(\eta,x) = E\, w(\theta,x)\ \underline{v}^T I(\theta,\eta)^{-1} \underline{v} - 2 \quad (4)$$

and

$$d_2(\eta,x) = E\ w(\theta,x)\ \underline{v}^T\, I(\theta,\eta)^{-1} A(\theta)\, I(\theta,\eta)^{-1}\, \underline{v} + \varphi_2(\eta). \quad (5)$$

For a design η_0 *to be* φ_1*-optimal the function* $d_1(\eta_0,x)$ *must be non-positive for all* x *in* χ *and for* η_0 *to be* φ_2*-optimal for a particular choice of* $A(\theta)$ *the function* $d_2(\eta,x)$ *must be non-positive for all* x.

4.1 THE DERIVATIVE FOR φ_1

We demonstrate using MACSYMA to find the criterion $\varphi_1(\eta)$ and the derivative $d_1(\eta,x)$. A record of using MACSYMA to do this is given as Figure 2. The matrices i and iinv are read in using the loadfile command, reading in from the file created in Figure 1. It is seen in expression (d2) that the criterion, the expected value of the log determinant of the information matrix, can be expressed as:

$$\varphi_1(\eta) = E \log(\beta^2 ts) .$$

Only the part of the derivative that is multiplied by $w(x,\theta)$ and then integrated numerically is calculated as expression (d4), that is $\underline{v}^T I(\theta,\eta)^{-1} \underline{v}$. The expand command simplifies the resulting expression and, recognizing the expansion of $(xbar-x)^2$, the derivative is:

$$d_1(\eta,x) = E\,[w(x,\theta)\,\{1/t + (xbar-x)^2/s\}] - 2.$$

The matrices i and iinv and the vector v are saved in file2 for use later.

```
% macsyma
This is UNIX MACSYMA Release 309.1.
(c) 1976,1984 Massachusetts Institute of Technology.
All Rights Reserved.
Enhancements (c) 1984 Symbolics, Inc. All Rights Reserved.
Type describe(trade_secret); to see Trade Secret notice.
Type exec("man macsyma"); for help.

(c1) i:entermatrix(2,2);

Is the matrix  1. Diagonal  2. Symmetric  3. Antisymmetric  4. General
Answer 1, 2, 3 or 4
2;

Row 1 Column 1:  b^2*t;
Row 1 Column 2:  -b*t*(xbar-m);
Row 2 Column 2:  s+t*(xbar-m)^2;
Matrix entered.
                    [          2                          ]
                    [         b  t        - b t (xbar - m)  ]
(d1)                [                                      ]
                    [                                  2   ]
                    [ - b t (xbar - m)   t (xbar - m)  + s ]

(c2) iinv:factor(invert(i));

Batching the file /usr/macsyma.309/share/invert.mac
Batching done.
                    [       2                  2                ]
                    [ t xbar  - 2 m t xbar + m  t + s  xbar - m ]
                    [ ------------------------------  -------- ]
                    [                2                   b s    ]
(d2)                [               b  s t                      ]
                    [                                           ]
                    [             xbar - m                1     ]
                    [             --------                -     ]
                    [               b s                   s     ]

(c3) expand(i.iinv);
                                  [ 1  0 ]
(d3)                              [      ]
                                  [ 0  1 ]

(c4) save([file1],i,iinv);

(d4)                          [file1, i, iinv]

(c5) quit();
```

FIGURE 1

```
(c1) loadfile(file1);
file1 being loaded.

(d1)                                done

(c2) expand(determinant(i));
                                     2
(d2)                                b  s t

(c3) v:matrix([-b],[x-m]);
                                  [  - b  ]
(d3)                              [       ]
                                  [ x - m ]

(c4) expand(transpose(v).iinv.v);
                                    2                2
                                 xbar    2 x xbar    x     1
(d4)                             ----- - -------- + -- + -
                                   s         s       s     t

(c5) save([file2],i,iinv,v);

(d5)                          [file2, i, iinv, v]

(c6) quit();
```

FIGURE 2

4.2 THE DERIVATIVE FOR φ_2

The weighted trace criterion of φ_2-optimality corresponds, approximately, to squared error loss of estimation. The criterion therefore requires that the quantities to be estimated, or predicted, are carefullly specified by the experimenter.

If, for example, the only parameter of interest is μ, then $A(\theta)$ can be written as $\underline{c}\,\underline{c}^T$ with $\underline{c}^T = (1,0)^T$. If both μ and β are of equal interest then $A(\theta)$ is the identity matrix. If the only quantities of interest are linear combinations of μ and β then the matrix $A(\theta)$ will not depend on the unknown parameters.

Alternatively, if a nonlinear function of μ and β is of interest then the matrix $A(\theta)$ will depend on the unknown parameters. For example it is often of interest to estimate the value of x at which the probability of success, $p(x,\theta)$, is a particular value. Suppose that we want to estimate x_0 where $\text{logit}\{p(x_0,\theta)\} = \gamma$, then $x_0 = \mu + \gamma\beta^{-1}$, which is a nonlinear function. Standard asymptotic arguments give $A(\theta) = c(\theta)\, c(\theta)^T$, where $c(\theta)$ is a vector of derivatives, $c(\theta) = (1, -\gamma\beta^{-2})$.

A distribution could be put on γ to represent interest in estimating several percentile response points x_0. This is a standard way of using the weighted trace criterion in a Bayesian framework.

Figure 3 shows how functions can be created in MACSYMA that calculates $\varphi_2(\eta)$ and $d_2(x,\eta)$ for any choice of $A(\theta)$. These functions are called criterion and deriv respectively and their use is illustrated, in Figure 3, for finding expressions for $\varphi_2(\eta)$ and $d_2(x,\theta)$ when μ is the only parameter of interest. These functions are created in (c2) and (c3). The matrix $A(\theta)$ for estimating μ alone is entered and used to find $\varphi_2(\eta)$ and $d_2(x,\theta)$ in (d5) and (d7) respectively. The dispfun command is used to display all user defined functions in (c8) and then in (c9) the functions are saved in a file called file3.

4.3 EXAMPLES OF φ_2-OPTIMALITY

The example in Figure 3 is the criterion where we suppose that interest is in estimation of μ alone. Then, as discusses earlier, $A(\theta)$ is $\underline{c}\,\underline{c}^T$ with $\underline{c} = (1,0)^T$. As shown in the MACSYMA output we have:

$$\varphi_2(\eta) = -E\,[\beta^{-2}\{t^{-1} + (\text{xbar} - \mu)^2 s^{-1}\}]$$

and

$$d_2(\eta,x) = E\,[w(x,\theta)\,(\beta st)^{-2}\{t(\text{xbar}-x)(\text{xbar}-\mu) + s\}^2] + \varphi_2(\eta).$$

Suppose alternatively that we want to estimate $x_0 = \mu + \gamma\beta^{-1}$ for a known value of γ. For example in engineering and reliability experiments it is sometimes of interest to estimate an extreme percentile response point, such as the point at which $p(x,\theta) = 0.95$. In this case the functions defined in MACSYMA could be used, with an appropriate choice of $A(\theta)$, to show that the criterion and derivative can be expressed as:

$$\varphi_2(\eta) = -E\,[-\beta^{-2}\{t^{-1} + (\gamma - \beta(\text{xbar}-\mu))^2\beta^{-2}s^{-1}\}]$$

and

$$d_2(\eta,x) = E\,[w(x,\theta)\,(\beta^2 st)^{-2}\{t(\text{xbar}-x)(\beta(\text{xbar}-\mu)-\gamma) + \beta s\}^2] + \varphi_2(\eta).$$

Finally suppose that we are interested in estimating several percentile response points. We can put a distribution over γ to represent this interest, as some points can be of more interest than others. Then the matrix $A(\theta)$ becomes:

$$A(\theta) = \begin{pmatrix} 1 & -E(\gamma)/\beta^2 \\ -E(\gamma)/\beta^2 & E(\gamma^2)/\beta^4 \end{pmatrix}.$$

For illustration, suppose we put a uniform distribution over $[-1,1]$ on γ. This represents an interest in calibrating the central part of the response curve. Then $E(\gamma) = 0$ and $E(\gamma^2) = 1/3$. The use of the two MACSYMA functions easily leads to the following expressions:

$$\varphi_2(\eta) = -E\,[\beta^{-2}\{t^{-1} + (\text{xbar}-\mu)^2/s + (3\beta^2 s)^{-1}\}]$$

and

$$d_2(\eta,x) = E[w(x,\theta)(\beta^2 st)^{-2} \times \{\beta^2(t(\text{xbar}-x)(\text{xbar}-\mu) + s)^2 + t^2(\text{xbar}-x)^2/3\}] + \varphi_2(\eta).$$

```
(c1) loadfile(file2);
file2 being loaded.

(d1)                                  done

(c2) criterion(a) := (a.iinv)[1,1] + (a.iinv)[2,2];

(d2)            criterion(a) := (a . iinv)      + (a . iinv)
                                          1, 1              2, 2

(c3) deriv(a) := transpose(v).(iinv.a.iinv).v;

(d3)         deriv(a) := transpose(v) . ((iinv . (a . iinv)) . v)

(c4) a:matrix([1,0],[0,0]);
                                     [ 1  0 ]
(d4)                                 [      ]
                                     [ 0  0 ]

(c5) criterion(a);
                                2                  2
                           t xbar  - 2 m t xbar + m  t + s
(d5)                       -------------------------------
                                        2
                                   b  s t

(c6) deriv(a);
                                        2
                    (x - m) (xbar - m)
(d6) (x - m) (------------------------
                          2  2
                         b  s

                          2                    2
   (xbar - m) (t xbar  - 2 m t xbar + m  t + s)
 - ---------------------------------------------)
                          2  2
                         b  s  t

                                    2                  2
     (x - m) (xbar - m) (t xbar  - 2 m t xbar + m  t + s)
 - b (-------------------------------------------------------
                                3  2
                               b  s  t

          2                    2       2
   (t xbar  - 2 m t xbar + m  t + s)
 - ---------------------------------)
                  3  2  2
                 b  s  t

(c7) factor(expand(d6));

                         2                                      2
                  (t xbar  - t x xbar - m t xbar + m t x + s)
(d7)              ------------------------------------------
                                    2  2  2
                                   b  s  t
(c8) dispfun(all);

(e8)            criterion(a) := (a . iinv)      + (a . iinv)
                                          1, 1              2, 2


(e9)         deriv(a) := transpose(v) . ((iinv . (a . iinv)) . v)


(d9)                                  done

(c10) save([file3],i,iinv,v,criterion,deriv);

(d10)                 [file3, i, iinv, v, criterion, deriv]

(c11) quit();
```

FIGURE 3

5. DISCUSSION

I have also used MACSYMA in studying optimal designs for other problems. Chaloner (1988) describes using MACSYMA to examine the problem of designing an experiment to estimate the point, in a quadratic regression, at which the response is maximized or minimized. This problem is studied in, for example, Buonaccorsi and Iyer (1986) and relevant results are given in Murty and Studden (1972). Both Bayesian and locally optimal designs are found and described in Chaloner (1987a) and the use of MACSYMA for the proof of these results are described in Chaloner (1988). Aspects of MACSYMA that are used in this problem include: finding a generalized inverse of a singular matrix, finding the roots of a quartic polynomial, taking derivatives of functions and plotting functions with symbolic arguments. Other features of MACSYMA that I have used in design and other problems are: writing a FORTRAN expression for inclusion in a program, taking a Taylor series expansion, finding integrals and taking limits.

MACSYMA is clearly a useful tool for these kinds of algebraic manipulations. Although I have not solved any problems that could I not otherwise have solved by careful, time consuming hand calculations, I have found MACSYMA extremely useful, fast and accurate. I believe that the initial effort in learning how to use MACSYMA to its fullest capabilities is well worth it. It is also fun to use.

REFERENCES

Buonaccorsi, J.P. and Iyer, H.K. (1986). Optimal designs for ratios of linear combinations in the general linear model. *Journal of Statistical Planning and Inference* 13, 345-356.

Chaloner, K. (1987a). Optimal Bayesian design for quadratic regression. *University of Minnesota, School of Statistics, Technical Report.*

Chaloner, K. (1987b). An approach to design for generalized linear models. In *Model Oriented Data Analysis*, V.V. Fedorov and H. Lauter eds.. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 3-12.

Chaloner, K. (1988). Using MACSYMA in optimal design. *Proceedings of the Statistical Computing Section of the American Statistical Association, 1987*, 22-30.

Chaloner, K. and Larntz, K. (1988). Optimal Bayesian design applied to logistic regression. To appear in *The Journal of Statistical Planning and Inference.*

Drinkard, R.D. Jr., and Sulinski, N.K. (1981). *MACSYMA: A program for computer algebraic manipulation (demonstrations and analysis).* Naval Underwater Systems Center, New London, Connecticut, Technical Document 6401.

Gong, G. (1983). Letting MACSYMA help. *Computer Science and Statistics: Proceedings of the 15th Symposium on the Interface*, edited J.E. Gentle, North-Holland, 237-244.

Murty, V.N. and Studden, W.J. (1972). Optimal designs for estimating the slope of a polynomial regression. *Jour. Amer. Statist. Assoc.* 67, 869-873.

Rand, R.H. (1984). *Computer Algebra in Applied Mathematics: An Introduction to MACSYMA.* Research notes in mathematics number 94, Pitman.

Rand, R.H. (1988). Computer algebra applications using MACSYMA. *Computer Science and Statistics: Proceedings of the 19th Symposium on the Interface*, 231-236.

Steele, J.M. (1985). MACSYMA as a tool for statisticians. *Proceedings of the Statistical Computing Section of the American Statistical Association*, 1-4.

Steele, J.M. (1988). An Application of Symbolic Computation to a Gibbs Measure Model. *Computer Science and Statistics: Proceedings of the 19th Symposium on the Interface*, 237-240.

ACKNOWLEDGEMENTS

Research for this paper was supported by the National Science Foundation under Grant No. DMS-8706754. I am grateful to L. Tierney and R. Weiss for their reading of this manuscript and helpful suggestions.

MACSYMA was developed at the Massachusetts Institute of Technology Laboratory for Computer Science and supported from 1975 to 1983 by the National Aeronautics and Space Administration under grant NSG 1323, by the Office of Naval Research under grant N00014-77-C-0641, by the U.S. Department of Energy under grant ET-78-C-02-4687, and by the U.S. Air Force under grant F49620-79-C-020 and since 1982 by Symbolics, Inc.. MACSYMA is a trademark of Symbolics Inc., Eleven Cambridge Center, Cambridge, MA 02142.

AN INTRODUCTION TO CART™: CLASSIFICATION AND REGRESSION TREES

Gerard T. LaVarnway, Norwich University

1. Introduction

The use of binary trees to perform classification provides an interesting alternative to classical parametric methods. CART™ is a fascinating mathematical theory that was developed by Leo Breiman (UC Berkeley), Jerome H. Friedman (Stanford), Richard A. Olshen (UC San Diego) and Charles J. Stone (UC Berkeley/UCLA) culminating in the monograph ***CART: Classification and Regression Trees*** (Breiman, et. al.) . In addition, CART™ was developed into a powerful software package that applies a nonparametric approach to classification and regression problems. Specifically, the software arrives at prediction rules in the form of binary decision trees.

This paper will discuss CART™ methodology as a tool in analyzing/solving classification problems. In addition, CART™ performs regression. However, this paper will focus solely on the classification problem. The procedure which performs regression is similar, but slightly different.

An appendix is provided which contains the complete CART™ processing and output on data from a classification problem. The entire CART™ output is provided for completeness.

2. Statement of the Problem

The general classification problem may be described as follows: Given a multivariate observation z which is known to belong to (emanate from) one of n possible populations (platforms), determine which population is most likely. The analyst who is performing this classification has an historic data base of observations, for each of which the actual population is known, and has suspicions - in the form of prior probabilities - regarding the likely population of z.

For clarity, let us define our measurement vector $\mathbf{x}$ to be an N-dimensional vector $\mathbf{x} = (x_1, x_2, x_3, \ldots x_n)$. CART™ allows for the variables x_n, to be of continuous and/or categorical type. A continuous variable is a variable that takes on real numbered values. A categorical variable is a variable that assumes a value from a discrete set, (e.g. {red, blue, green}). The vector $\mathbf{x}$ is known to belong to one of j classes j = 1, 2, ... J.

In performing classification, an analyst records the observation vector, $\mathbf{x}$, of an object and predicts the class, j, to which the object belongs.

Sample classification problems are as follows:

- At the University of California at San Diego (UCSD) Medical Center, incoming heart attack patients are monitored on 17 different variables (blood pressure, age, etc.). The medical staff would like to predict if the patient is in a high or low risk of death (Breiman, et al. 1984).
- Determine a ships class (destroyer, cruiser, submarine, battleship, aircraft carrier, etc.) from surveillance observations.
- Predict a college freshman's success or failure in his/her first mathematics course from various previous test measurements (e.g. SAT scores, etc.).

3. CART™ METHODOLOGY

This section provides a brief summary of CART™ processing. For a complete description of CART™ and its supporting theory, the reader should consult the monograph, Breiman, et. al. (1984).

CART™ arrives at its classification rule(s) by producing a binary decision tree which partitions a set into disjoint subsets. This partition has the property, that for any element of a given subset, a class can be assigned.

A sample classification tree might look as follows:

Sample Classification Tree

figure 2-1

To construct a binary decision tree, a learning sample is required. A learning sample is a set of measurement observations, for which the true class of each measurement observation is known. It is desired to include in the measurement vector, all variables which are believed to have some predictive power in determining the classification of the measurements. CART™ uses the learning sample to construct a decision tree that can then be used to classify an observation whose class is unknown.

In the sample decision tree (figure 2-1), we observe CART™'s partitioning of the space into descendant nodes (subsets). The square boxes indicated terminal nodes. A terminal node is a node at which a class assignment can be made. The circular nodes indicate nonterminal nodes, where a class determination cannot be made. At each nonterminal node, a binary (yes/no) question is asked, "splitting" that node into two descendant sub - nodes, which may or may not be terminal nodes.

The concept of splitting is fundamental to CART™ processing. CART™ allows for three different types of splits:

1) univariate splits on a continuous variable: is $x_n \leq C$, C a fixed real number.

2) linear combination splits on continuous variables:is $c_i x_i + c_j x_j \ldots + C_n x_n \leq C$, C a fixed real number

3) splits on a categorical variable: is x_n is an element of a finite set $\mathbf{S}$, then ask the question, "Is $x_n \in S$", where S ranges over all possible subsets of $\mathbf{S}$.

Another natural question is "How does CART™ decide on a particular split for a given node?" The choice of a split is made on the notion of impurity. CART™ chooses the split that minimizes the impurity. "What is meant by impurity?"

Definition 3.1: Call $\phi = \phi(p_1, p_2, \ldots p_k)$, a function of non-negative arguments with $\sum_{j=1}^{k} p_j = 1$, an impurity function if

1) $\phi \geq 0$

2) ϕ is maximum when $p_1 = p_2 = \ldots p_k$

3) $\phi = 0$ when $p_j = 1$ for some j

where k is equal to the number of classes.

Example 3.1: The entropy measure of impurity is given by

$$\phi = -\sum_{j=1}^{k} p_j \log p_j$$

With 0log0 = 0

Example 3.2: The Gini measure of impurity is given by

$$\phi = 1 - \sum_{j=1}^{k} p_j^2$$

Once an impurity function has been defined and selected, we ultimately define the impurity of a node and the impurity of a tree.

Definition 3.2: The impurity $i_\phi(t)$ of a node t is

$i_\phi(t) = \phi(p(1|t), p(2|t), \ldots p(k|t))$

where p(j|t) is the estimated probability of a class j object at node t.

Definition 3.3: The impurity $I_\phi(T)$ of a tree T is

$$I_\phi(T) = \sum_{t \in T} i_\phi(t) p(t)$$

where $\tilde{T}$ denotes the set of terminal nodes of T and p(t) is an estimate of the probability that a case falls into node t.

With the above definitions, CART™ selects the split that minimizes the overall tree impurity.

We now have established the procedure for splitting a nonterminal node into descendant subnodes. However, the issue of how CART™ selects the optimal decision tree for classification has not been addressed.

CART™ continues splitting (partitioning) until an overly large tree, T_{max} is grown. That is, a tree with all terminal nodes pure or have a count less than or equal to some small number (default 5). A process known as "pruning" generates a nested sequence of subtrees. A subtree is created by pruning off a branch or branches from the previous subtree. Selection of the branch to be pruned is done by a cost complexity measure.

The cost complexity measure is a measure of the resubstitution estimate of misclassification and a "penalty" for the complexity of the tree.

Definition 3.4: For a given tree T let $M_\alpha(T)$,

$$M_\alpha(T) = R(T) + \alpha|\tilde{T}|,$$

be the cost complexity of T with complexity parameter α, $\alpha \geq 0$. R(T) is the resubstitution estimate for misclassification cost of tree T. $|\tilde{T}|$ is the number of terminal nodes in tree T.

We see from the above definition that by increasing the value of α, we increase the penalty for the complexity of the tree.

By the pruning technique, CART™ generates a sequence of nested subtrees $T_1, T_2, \ldots T_{max}$. Associated with each tree in this sequence is an estimate of the misclassification cost for that tree. Three methods for estimating this misclassification cost are available: resubstitution, test sample, and cross validation. (see Breiman et al. 1984, pp 72-81).

Naturally, one would think that CART™ selects the subtree with the minimum misclassification cost. However, there is some uncertainty associated with the misclassification estimates. CART™ resolves this uncertainty by calculating their standard errors (SE) (see Breiman, et. al. 1984, pp 78 - 81).

CART™ then selects the subtree with the least number of terminal nodes, within one (1) standard error. The decision to select the subtree with the minimum number of terminal nodes, is due to the fact that a simpler tree is preferred. CART™ allows the user, as an option during execution, to vary the SE rule. For example, if the user desires the tree with the absolute minimum misclassification, set the SE rule to 0.0SE. Any variation of this SE rule is allowed, 2SE, 1.5SE, etc.

Once an optimal subtree has been selected, objects whose class is unknown may be passed down the decision tree for classification.

The final issue of importance is "How is the class assignment performed?" This is done in the most natural way. If the misclassification costs are equal, the assignment at each terminal node is the most populous class of the learning set in that node. If the misclassification costs are not equal, CART™ assigns class j* to node where j* minimizes

$$\sum C(j|m)p(m|t)$$

where C(j|m) is the cost for classifing a class m object as a class j object and p(m|t) is the probability of class m object at node t.

To summarize, CART™ constructs a binary decision tree in the following manner:

1) Produce an overly large tree using binary questions, minimizing tree impurity at each step.

2) Prune this large tree generating a nested sequence of subtrees, each with an associated misclassification cost.

3) Select the optimal tree for use as a classifier.

Any discussion of CART™ would not be complete without mentioning some of CART™'s nonstandard features, that make the software so attractive. A list of the nonstandard features that I find

useful are as follows:

1) CART™ is a nonparametric approach. It is nonparametric in the sense that it places no restrictions on the distribution(s) of any variable(s) (e.g. normality is not assumed).

2) More classical statistical methods cannot deal with missing data in a natural way. CART™ handles missing data by the use of "surrogate splits". When a split is selected, CART™ measures the association of splits on other variables to the chosen split. In the event data is missing for a split in the tree, CART™ would then split on the variable with the greatest association. This associated split is called a surrogate split.

3) Linear combination splits are allowed. If the structure for a given problem depended on a combination of variables, univariate splits would prove unsatisfactory. As mentioned earlier, CART™ allows for linear combination splits.

4) Variable importance: CART™ provides as part of its output a ranking of the variables. These may prove useful in identifying variables with the most predictive power.

SUMMARY AND CONCLUSION

The reader has been introduced into the classification problem and the use of binary tree classifiers. Specifically, CART™ has proven to be a procedure rich in mathematical theory, as well as, a powerful software package that performs classification and regression.

The many nonstandard features that CART™ supports makes it appealing. In addition to being a nonparametric approach, it also provides an interesting alternative to more classical statistical methods.

CART™ 's decision rules are easy to use, understand and interpret. It provides interesting analysis of problems from various disciplines including, the social sciences, medicine, physical science, surveillance, etc..

Bibliography

Breiman et al. Classification and Regression Trees. Belmont: Wadsworth, Inc., 1984

California Satistical Software Inc.. "CART™ Output" CART™ Version 1.1 October 1986.

Acknowledgements

My original research and introduction to CART™ occurred during a research appointment in the 1986 U.S. Navy - American Society for Engineering Education (ASEE) Summer Faculty Research Program. The research, conducted at the Naval Ocean Systems Center (NOSC), San Diego, CA 92152 was supported by the Tactical Information Correlation and Presentation (TICAP) project of the combat direction block program 62721N, subproject N022C RX-242-4431.

The assistance and support of Dr. Roger Johnson, Code 421, Naval Ocean Systems Center was greatly appreciated while conducting research at NOSC.

Permission to use selected portions of the monograph Classification and Regression Trees by Breiman, et al. (1984) was kindly granted by Carline Haga, Permissions Manager for Brooks/Cole and Brooks/Cole Advanced Books and Software.

CART™ FORTRAN source code Version 1.1 was purchased from California Statistical Software Inc., 961 Yorkshire Court, Lafayette, CA 94549.

Generating Code for Partial Derivatives: Some Principles and Applications to Statistics

John W. Sawyer, Jr.
Texas Tech

Abstract

The author re-examines his previous results on generating code for first partial derivatives of a function using the natural action of a compiler: Source code for the function alone yields object code for its first partials, and this object code executes in a time at most proportional to the execution time for the original function. (The proportionality constant does not depend on the function or the number of its arguments.) Implications of these results for the generation of code for higher order partials will be discussed, as will applications to some statistical methodologies.

1. Introduction

In Sawyer (1984) the author presented a strategy for computation of first partial derivatives of a function. This strategy can be integrated smoothly into the natural action of compiler. The development of this strategy was motivated by a desire to streamline the manner in which a function had to be specified for Grizzle, Starmer, and Koch (1969) analysis of categorical data, though the applications of the strategy are much wider. A brief discussion of applications of strategies for efficient, transparent computation of first and higher order partials will be found at the end of this paper, though statisticians should require little convincing of the value of a convenient way to get efficiently computed derivatives.

Subsequent discussions with colleagues have led the author to conclude that there is some confusion about what is different in the Sawyer (1984) paper from other attacks on automatic differentiation such as the monograph by Rall (1981). The answer to this is, that, to the best of the author's knowledge, it had not been pointed out before that (i) without changing its scanner or parser, a compiler which is capable of producing object code to evaluate a user-programmed function can be modified to produce object code for first partials (no symbol mannipulation of source code is involved) (ii) this modification will produce object code which computes all partials in a time proportional to the time which it takes simply to evaluate the user function. It is not the intent of this paper to redevelop these ideas in detail, as they are discussed at length in Sawyer (1984). A brief review of the basis for (i) and (ii) above is useful, however, in that it suggests how relatively efficient automatic generation of higher order partials might proceed. The next section provides such a review, while Section 3 discusses second partial generation.

2. First Partial Generation: A Review

Consider the arbitrary function programmed in a high level language such as FORTRAN:

```
F=((X2**2)*COS(X1))/ (X1+2.*EXP(X1/X2))
```
(2.1)

A compiler will first scan this code to identify variables, constants, operators, and relations, translating each into appropriate numeric codes. This numeric translation will then be parsed, resulting typically in a object code which is represented here in high-level form for reader convenience:

```
A1=X2; B1=A1**2; A2=X1; B2=COS(A2);
C1=B1*B2; A3=X1; A4=2.; A5=X1; A6=X2;

B3=A5/A6; C2=EXP(B3); D1=A4*C2;
E1=A3+D1; F=C1/E1;
```
(2.2)

The trivial statements "A1=X2", "A2=X1", etc., represent the recognition by the parser of constant or variable. Some of the statements in (2.2), such as "E1=A3+D1", can be carried out readily in an arithmetic register, while a statement such as C2=EXP(B3) will require a macro of some sort. The important point, is, however, that as long as one is working with floating point numbers with mantissa and exponent of a fixed maximum number of significant digits, there will be an upper and lower bound on the time needed to execute each of the primitive steps in (2.2).

Let us associate algebraic variables x_1 with X1, a_1 with A1, etc. Then Figure 1 gives a parse tree for the function (2.1) in terms of these algebraic variables. The purpose of labeling the edges of the tree with partials as shown becomes clear when we note that

$$\frac{\partial f}{\partial x_2} = \frac{\partial b_1}{\partial a_1}\frac{\partial c_1}{\partial b_1}\frac{\partial f}{\partial c_1} + \frac{\partial b_2}{\partial a_2}\frac{\partial b_1}{\partial b_2}\frac{\partial d_1}{\partial b_1}\frac{\partial c_1}{\partial d_1}\frac{\partial f}{\partial c_1} \qquad (2.3)$$

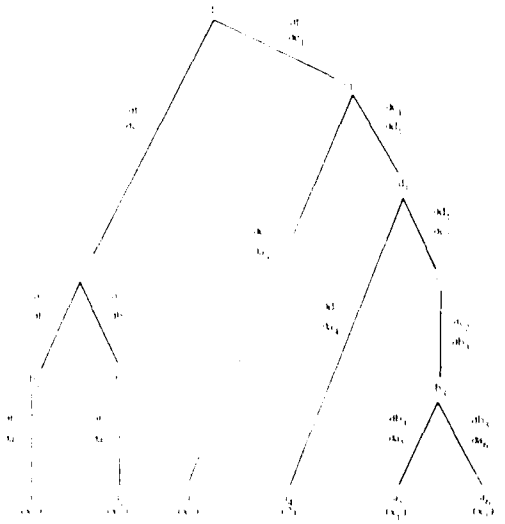


FIGURE 1

As is discussed in Sawyer (1984), this derivative can be computed by simply multiplying the partials found along the edges from the root to the leaves of the tree corresponding to x_2 and summing up the products.

Now what is the payoff in the "multiplying-down-the tree" trick from the point of view of object code? Why do we not simply, in effect, choose each leaf in left to right sequence, multiply up the tree, and add to accumulators for the partials? The answer is that one must be careful not to do redundant multiplications. The strategy of doing multiplications from the top down leads naturally to a code generation scheme which does avoid redundant multiplications.

Code to compute first partials of a function being parsed such as (2.1) may be generated as follows. When the parser recognizes a node, code to evaluate the partials of that node respect to its arguments is generated at the same time as code to evaluate the node. For instance, after the node C2 is recognized in the process of parsing (2.1), the code "DC2DB3=C2" will be generated as well as the statement "C2=EXP(B3)" already seen in (2.2) above. ("DC2DB3" is simply a reader convenience representing a storage location in which the partial of C2 with respect to B3 is to be kept.) A statement of the form "DFDB3 = DFDC2 * DC2DB2" will also be generated and pushed onto a special stack for such statements. After an entire function such as (2.1) is parsed and other object code generated, the elements of this stack will be popped, causing them to enter the stream of object code in reverse order. A portion of this code popped from the stack for (2.1) would look like this:

```
DFDD1 = DFDE1*DE1DD1; DFDA3 = DFDE1 *
DE1DA3; DFDX1= DFDX1+DFDA3; DFDC2 =
DFDD1*DD1DC2; DFDB3 = DFDC2 * DC2DB3;
DFDA6 = DFDB3 * DB3DA6; DFDX2 =
DFDX2+DFDA6; DFDA5 = DFDB3 * DB3DA5;
DFDX1 = DFDX1 + DFDA5; ... (2.4)
```

Note that when (2.4) is actually executed that, according to the the way the object code for (2.1) has been generated, every variable on the right hand side of any statement in (2.4) is already well defined. (Assume accumulators have been zeroed.) Note also that DFDB3 is computed only once, and suffices to compute both DFDA6 and DFDA5 in two more steps. Thus, while we have, in effect, "multiplied down the tree" to both the leaves A5 and A6, we have not performed any redundant multiplications.

A little thought shows that the above approach yields object code which performs a number of multiplications "down the tree" which is less than the number of edges in the tree. (The number of such multiplications is in fact exactly equal to the number of edges in the tree less the number of arguments of the root node.) Further, in the case of nodes which represent binary or unary operations, as is the case for (2.1), the time taken to evaluate all the partials along the edges of Figure 1 must be bounded by a time proportional to the number of edges in the tree. (There must be some maximum time which it takes to compute a partial of any of a finite set of built-in unary or binary operators). Finally, the time it takes to add the partials of the function with respect to its leaves to the appropriate accumulator is certainly bounded by a time proportional to the number of leaves in the tree. Since, as we have already noted above, there is a maximum time which it takes to evaluate any of a finite set of unary or binary built-in operators, it follows that the time it takes simply to evaluate a function such as (2.1) is also proportional to the number of edges in the tree. Hence, since both the times needed to evaluate the function and the to evaluate all its partials are proportional to the number of edges in the parse tree, it follows that the partials of such functions can be evaluated in a time at most kt, where t is the time in which the function is evaluated and k depends only on the high level language and machine used.

The reasoning above can also be used to apply the kt result to functions involving sum and product operators, though, as Sawyer(1984) points out, care must be taken so that computation time for product operators remains proportional to the number of edges in the function. The reader is referred to that paper for details.

It should be noted that the kt bound is attained for one relatively simple function: the product of n distinct variables. The efficient way to calculate its partial with respect to a given variable is to divide the product by that variable (assuming none of the n variables are 0 when the product is evaluated). Both the time needed to calculate the product and the time needed to calculate the partials by division will be proportional to n for large n. Further, k should roughly be the ratio of the time it takes to do a floating point division to that which it takes to do a floating point multiplication.

3. Toward Automatic Generation of Efficient Code for Second Partials

What follows is a brief verbal description of how automatic generation of efficient code for second partials of a user-programmed function might be carried out. The attack is a extension of the paradigm above and of Sawyer (1984). (An exhaustive treatment of this topic a bit involved to be presented in a few Proceedings pages.)

We first note that there is a lower limit on how efficient computation of second partials of a function can be, compared to the time it takes simply to evaluate the function. Return to the example of the product of n distinct variables discussed above. The second partial of the product with respect to two of these n variables may be computed by dividing the product of all the variables by the product of the two variables in question. This means that, for a specified set of values of the variables, n(n-3) operations are required to obtain the partials. (This figure arises from the fact that two operations are required to compute a partial with respect to each combination of variables, and that the second partial of a product with respect to the same variable taken twice is 0). Since the time needed to compute the product is proportional to n, it follows that the time needed to compute the second partials as above should be proportional to the square of the function evaluation time.

The reader should satisfy him- or herself that a bound proportional to the square of product evaluation time cannot be improved. Remember, each distinct variable is free in general to take on any value. In particular, the respective values can be the first n primes. The (n(n-1)/2)-n non-zero second partials evaluated for these values of the variables will all be distinct numbers. Since the number of distinct partials to be evaluated is proportional to the square of the product evaluation time for large n, and since each of these partials must involve at least one arithmetic operation apiece to produce it, the proportional-to-time-squared bound cannot be beaten for products. (We can, of course, do much worse. Each second partial of the product with respect to two distinct variables is defined as the product of the remaining n-2 variables. If we compute all non-zero partials in this manner, we perform (n-3)n(n-1)/2 multiplications, so that our bound on second partial computation time becomes proportional to product computation time cubed.)

In principle, it should be possible to compute the second partials of a function in a time proportional to the square of the evaluation time for the function. Consider a function h which is a can be programmed as a composite of built-in and user supplied functions (for which the user also supplies first and second partials). Suppose we write h as

$$h=h[g_1(f_1,\ldots f_n),\ldots g_m(f_1,\ldots f_n)] \quad (3.1)$$

Now let us specify that for a user program which evaluates h that a compiler will recognize the g's as either built-in or user specified functions with f's as arguments, that is, on recognizing the g's it will immediately be able to produce code for the first and second partials of the g's with respect to the f's. Formally, the f's must be daughters of the g's in the parse tree with root h. On the other hand, we make no restriction on how deep the g's may be in the parse tree. We ultimately want second partials of h with respect to a certain set of variables; each of the f's will be roots of parse trees which have these variables and constants for leaves.

Now by a double application of the chain rule we get

$$\frac{d^2 h}{df_i\, df_j} = \left\{\sum_s \sum_t \frac{d^2 h}{dg_s\, dg_t}\,(dg_s/df_i)(dg_t/df_j)\right\} + \left\{\sum_s (dh/dg_s)\,\frac{d^2 g_s}{df_i\, df_j}\right\} \quad (3.2)$$

There is actually quite a bit in the structure of (3.2) that we can take

advantage of for efficienct code generation, with a bit of massaging. Note that there is a recursive structure to (3.2): given that we have the first and second partials of h with respect to the g's, and that we can generate code to compute the first and second partials of the g's with respect to the f's, we do indeed get the partials of h with respect to the f's. Repeated application of (3.2) will, not surprisingly, take us from the root of the parse tree down to a point at which the partials of h with respect to the variables of interest will be obtained. Yet how we apply (3.2) in such a way as to insure that the partials will be obtained in a time proportional to function execution time squared is not immediately apparent.

The trick for obtaining the kt bound on first partial computation is to keep the computation time for those partials proportional to the number of edges in the parse tree. Similarly, what we want to do for second partials is to keep the execution time for second partial computation proportional to the number of edges in the tree squared. One way to do this is to traverse the tree in a outer loop, from right to left, stopping at each leaf to traverse the tree from that leaf to the right. If this this procedure computes the second partials in such a way that there is some fixed, maximum number of operations of maximum time duration for each edge traversed, then the second partials will be computed in a time proportional to function execution time squared.

In fact, such a nesting of tree traversals can be accomplished. In practice most of the terms in (3.2) drop out as one moves from the root to the leaves of the parse tree. For instance, consider the contribution of the variables A1 and A6 seen in (2.2) and the parse tree Figure 1 to an accumulator for the second partial of F with respect to X1 and X2. In this instance (3.2) will reduce to twice the second partial of F with respect to C1 and E1 times the product of first partials down the edges from C1 to A1 times the product of the first partials down the edges from E1 to A6. As another example, to get the contribution of the nodes A5 and A6 to the appropriate accumulator, one applies (3.2) iteratively to F[E1(D1)], F[D1(C2)], and F[C2(B3)], and then multiplies the second partial of F with respect to B3 by the first partials associated with the arguments of B3. Finally, the second partial of B3 with respect to A5 and A6 is multiplied by the product of first partials down the edges from the root to B3, and the result added to the foregoing product. For each iteration (3.2) in this instance, each of the two sums on the right hand side of the equation will involve only one term.

In general, the contribution of any two leaves in a parse tree to an accumulator for second partials will be computed as follows: through iterative application of (3.2), compute the second partial of the root with respect to that node at which the paths up the tree from the leaves in question join. Each iteration will involve only single terms for each of the two sums on the right hand side of (3.2). One more iteration is done to obtain the second partial of this node with respect to the arguments on the paths from this node to the leaves in question. The rest of the computation is then simply a matter of multiplying this second partial by the products of the first partials associated with the edges connecting these arguments with their respective leaves.

Now it should be evident that as the contributions of pairs of leaves to accumultors for second partials are collectively computed that many computations do not have to repeated for every pair. Once we have computed (from the top down) the first and second partials of the root with respect to a node in the tree, or the second partial of the root with respect to several arguments of a node within the tree, we do not have to recompute these partials every time we want the contribution of pairs of leaves which have paths up to these arguments. All we need once these things are computed are the products of the first partials down the edges from these arguments to the leaves. But the first partial computation process of Section 2 above provides the product of first partials down the edges from the root to the these arguments and from the roots to the leaves. Thus the product from the appropriate argument down to a leaf can be obtained by a single division. If we now require that the second partials of the operation represented by any node with respect to its arguments be computable in time proportional to the squared of the time it takes to evaluate the node, then all the forgoing will fit together to produce a strategy for evaluation of second partials for which the computation time is indeed proportional to the number of edges in the parse tree squared. Binary and unary operators will meet the requirement, as will sum and product operators, if the latter is properly handled.

Object code for second partials is generated along the lines that code for first partials is produced, including the use of a stack of code to be output in reverse order after the root is recognized. Each time a node in the parse tree is recognized, code for second partials with respect to its

arguments must be generated. Code from the stack relating to second partials mirrors the behavior of code in that stack for computation of first partials, although (3.2) must now be appropriately incorporated (Again, the exact particulars are beyond the scope of this paper). As should be evident from the foregoing discussion, considerable backtracking in the actual execution of the object code popped from the stack will be involved, as the contribution of the combination of each leaf and every to its right to some accumulator for second partials must be computed. This would appear to necessitate a system of pointers not necessary for first partial computation alone.

Problems still remain with the second partial computation scheme sketched above for which space does not allow for ample discussion. One problem is dangling nodes in the tree which are not really proper leaves. These correspond to variables in source code which appear on the left hand side of an assignment statement once and on the right hand side of a statement more than once. This problem is solved in Sawyer(1984) for the first partial case, and the second partial case is a generalization of that solution. Space complexity is an issue even with first partial code generation, and will be even more so with second partials. A certain amount of recomputation of partials, rather than storing them indefinitely, may be necessary for some functions as a proper tradeoff between space and time costs.

4. Applications

As stated above, the authors work on first partials was motivated by a desire to streamline weighted least squares analysis of categorical data. Some discussion of applications of the strategy of Section 2 for first partial code generation is given in Sawyer (1984), and to some extent this carries over to second partial generation as well.

To briefly carry the applications idea further, consider the case in which a likelihood is to be maximized over a large number of (say nuisance) variables. Even if the second partials matrix is not used in the search routine itself, that matrix may be preferable as a source of asymptotic variances of estimated parameters, or even as a criterion which can be checked for negative definiteness to verify that one is indeed at a maximum. Another application may be to biased estimates of a function of a large number of nuisance parameters, if such estimates are constructed from a set of reasonably consistent, unbiased estimates of these nuisance parameters. The first term in a Taylor series expansion of the bias in the estimator of the function will involve second partials of the function. Ability to compute these partials readily may allow the construction of a beneficial bias correction term.

References

GRIZZLE,J.E., STARMER, C.F., and KOCH, G.G. (1969), "Analysis of Categorical Data by Linear Models," Biometrics, 25, 489-504.

RALL, L.B. (1981), "Automatic Differentiation: Techniques and Applications," Lecture Notes in Computer Science, No. 120, New York: Springer-Verlag.

SAWYER, J.W., Jr. (1984), "First Partial Differentiation with an Application to Categorical Data Analysis", The American Statistician, 38, 300-308.

NOISE APPRECIATION: ANALYZING RESIDUALS USING RS/EXPLORE

David A. Burn and Fanny L. O'Brien, BBN Software Products Corporation

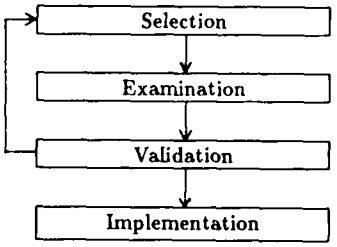
The RS/Explore software is a statistical advisory environment for performing analysis of general linear models. One goal of data analysis is to find a "model" that adequately describes the variation in the data. Residual analysis is an invaluable tool in selecting and validating a model. We will examine how RS/Explore provides convenient access to traditional and innovative graphical displays useful in residual analysis.

KEY WORDS: Residual Analysis, Studentized Residual, Influence, Leverage, Cook's Distance

1. INTRODUCTION

A primary objective of data analysis is to find a *model* that adequately describes the *variation* in the *data*. The process of building models consists of the following steps:

1. Model Selection
   (a) Determine general class of models.
   (b) Identify parsimonious subclass of models.
   (c) Apply transformations to data.
2. Model Examination
   (a) Fit model to data.
   (b) Compute estimates of parameters.
3. Model Validation
   (a) Check model assumptions.
   (b) Check model fit.
4. Model Implementation
   (a) Predict future values of the process.
   (b) Control future values of the process.



Approach to Model Building

2. RS/Explore Software

2.1 Objectives of the Software

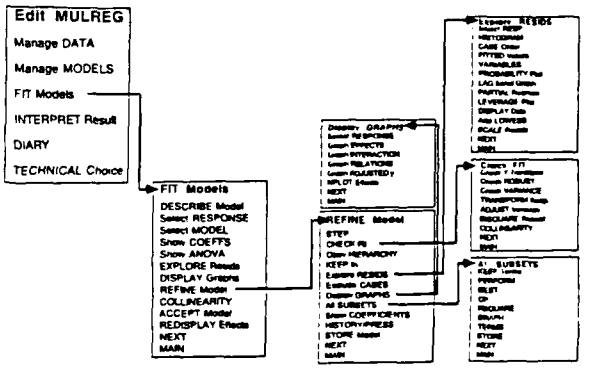
The RS/Explore software provides

- an interactive *computing environment* for data analysis, regression modelling, and interpretation of results
- a *menu system* which allows the program to be used effectively by nonstatisticians, especially industrial scientists and engineers
- *statistical tools* which help the data analyst avoid the most common pitfalls and inappropriate analyses in regression modelling

2.2 Menu System for Building Models

The menu system in RS/Explore encompasses the iterative approach to building models. In particular, the activity of model validation is simplified by the ability to *explore residuals* through a variety of traditional and innovative graphical displays.

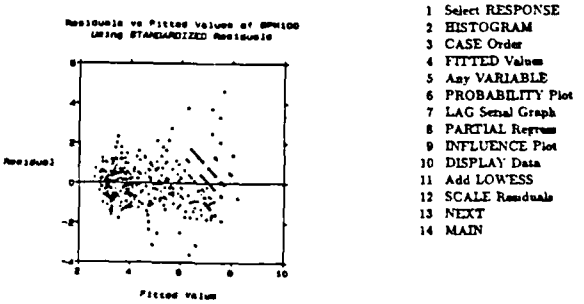
Menu System for Building Models



2.3 Screen Display for RS/Explore

The terminal screen in RS/Explore is partitioned into three regions: graphics, menu, and dialogue. In the *graphics region*, RS/Explore displays graphical objects such as boxplots and scatterplots, and nongraphical objects such as AOV tables and coefficients tables. In the *menu region*, RS/Explore displays the list of currently available options, and highlights one or more of these options as appropriate next steps in the data analysis. In the *dialogue region*, RS/Explore displays information regarding interpretation of statistical procedures and echos keyboard input.

Screen Display for Residual Analysis



2.4 Formulas for Regression Diagnostics

The RS/Explore software defines formulas for regression diagnostics as follows:

Residual

$$e_i = Y_i - \hat{Y}_i$$

Studentized Residual

$$e_{(i)} = \frac{e_i}{\left(s^2_{(i)}(1-h_i)\right)^{1/2}}$$

Mean Squared Error

$$s_i^2 = \frac{\sum_j e_j^2}{n-p}$$

Studentized Mean Squared Error

$$s^2_{(i)} = \frac{\sum_{j \neq i} e_j^2}{n-p-1}$$

Cook's Distance

$$C_i = \frac{h_i}{ps^2}\left(\frac{e_i}{1-h_i}\right)^2$$

Leverage Point Rule

Any observation such that $h_i > 2p/n$

Influence Point Rule

Any observation such that $C_i > C_{.95}$

3. Data Analysis Example

3.1 Description of Dataset

The Car Dataset consists of $n = 385$ observations on 8 characteristics of automobiles.

Name	Units	Scale
MPG	mi/gal	Measurement
CYLINDERS	4, 6, 8 cyl	Rank
DISPLACEMENT	cu in	Measurement
HORSEPOWER	hp	Measurement
WEIGHT	lbs	Measurement
SLUGGISHNESS	sec/(0.25mi)	Measurement
YEAR	1969–1981	Rank
ORIGIN	continent	Category

3.2 Identification of Model

Our objective is to determine the relationship between gasoline economy (response) and weight (predictor) and number of cylinders (predictor).

Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

Symbol	Variable
Y	GPM100 = 100/MPG
X_1	WT100 = WEIGHT/100
X_2	CYLINDERS, (C=6)–(C=4)
X_3	CYLINDERS, (C=8)–(C=4)

3.3 Least Squares Fit

The fitted model and the analysis of variance table are as follows:

Fitted Model

$$\hat{Y} = 1.0175 + 0.1304X_1 - 0.0403X_2 + 0.5210X_3$$

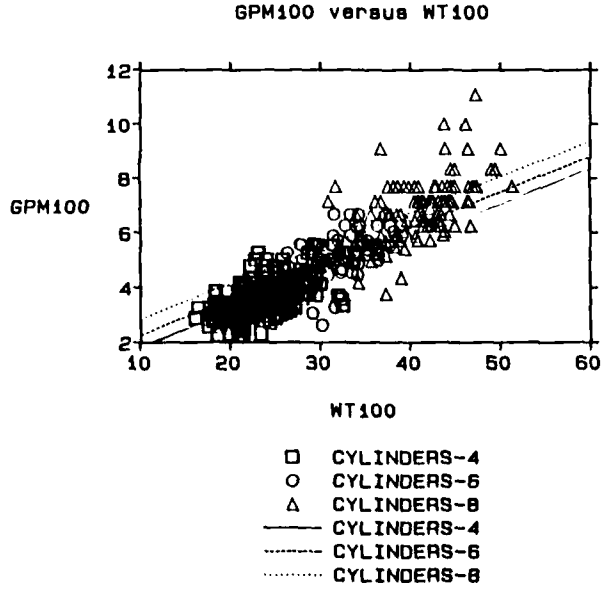
Analysis of Variance

Source	df	Sum of Squares	Mean Square	F-Ratio	p-value
Regression	3	870.42	290.14	536.40	0.0000
Residual	381	206.09	0.54		
Lack of fit	339	191.79	0.57	1.66	0.0232
Pure Error	42	14.29	0.34		
Total	384	1076.51			

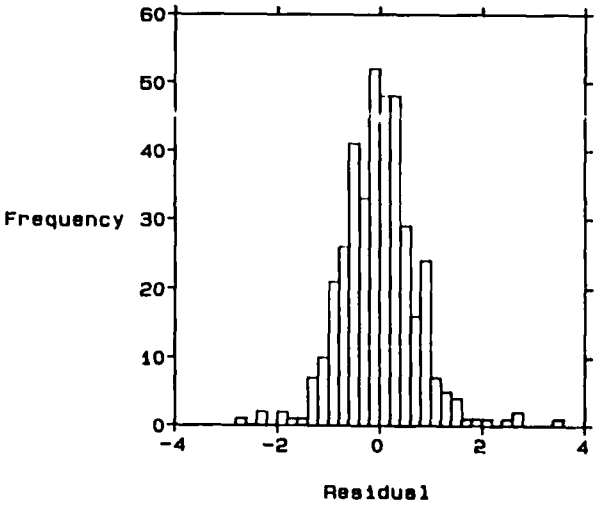
R-squared = 0.8086
Adjusted R-squared = 0.8070
Standard Error = 0.7355

3.4 Analysis of Residuals

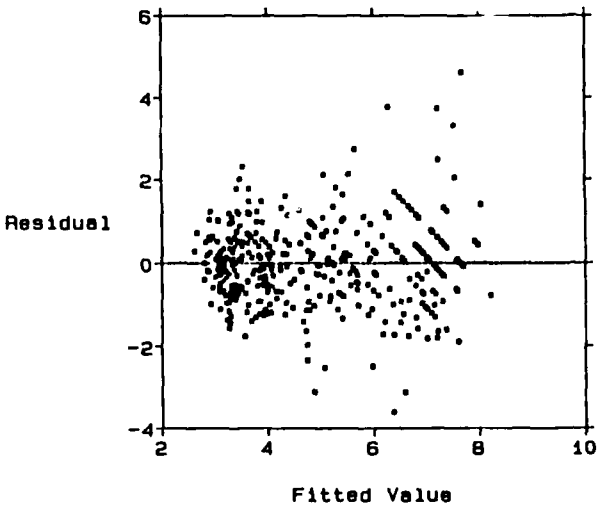
The residuals from the fitted model are examined using a variety of graphical displays. The scale of the residuals in all displays may be specified as raw, studentized (default), absolute raw, or absolute studentized. A lowess curve may be added to all residual displays to identify trend.



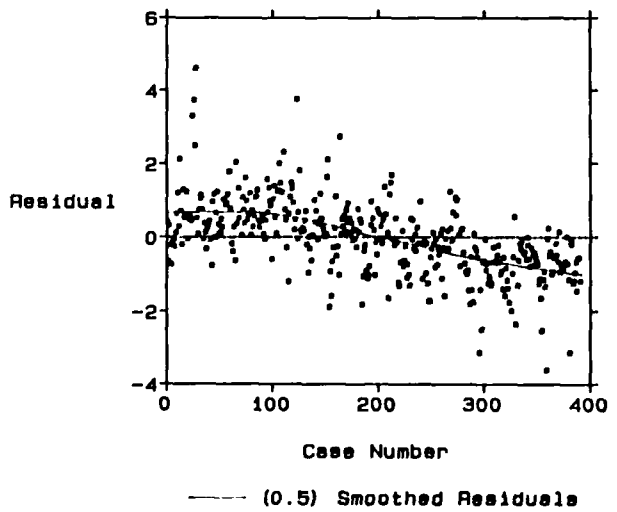
Histogram of Residuals
Using RAW Residuals



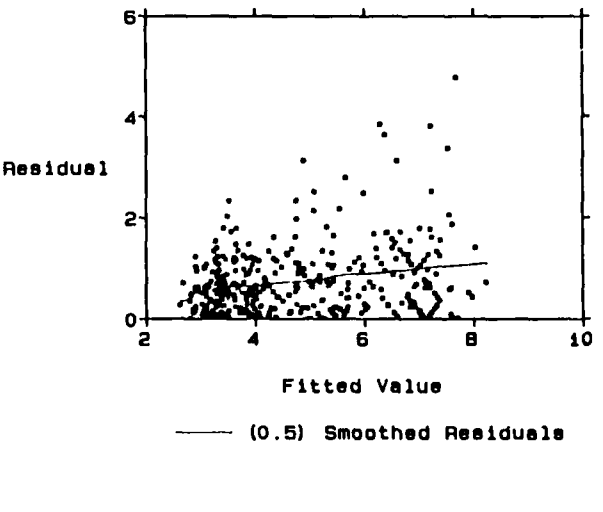
Residuals vs Fitted Values of GPM100
Using STANDARDIZED Residuals

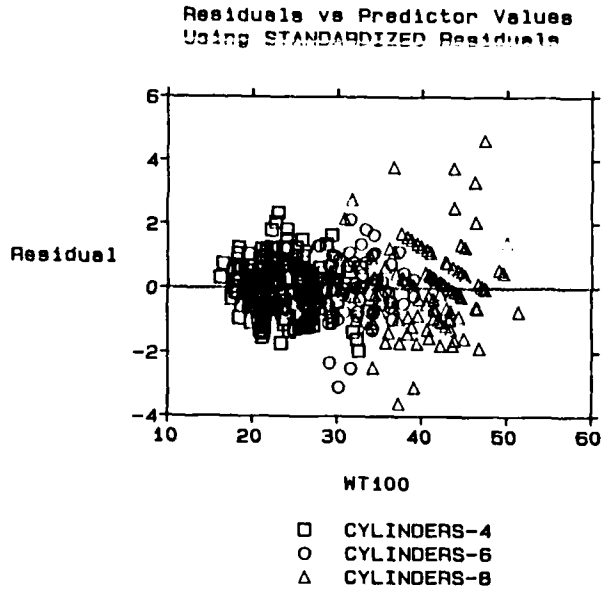


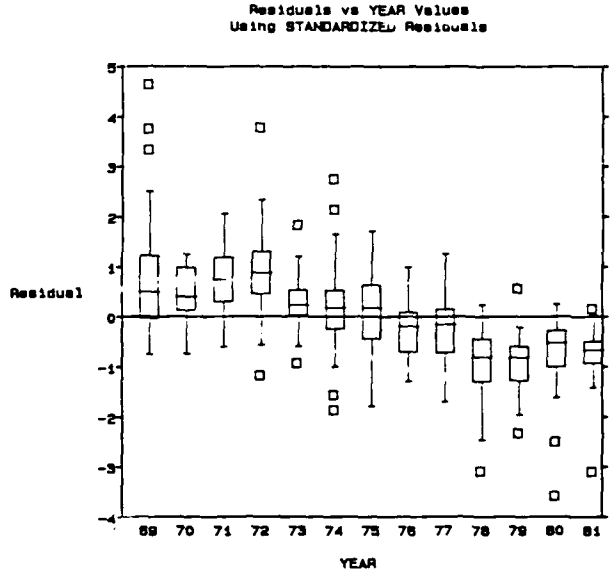
Case Order Graph of Residuals
Using STANDARDIZED Residuals

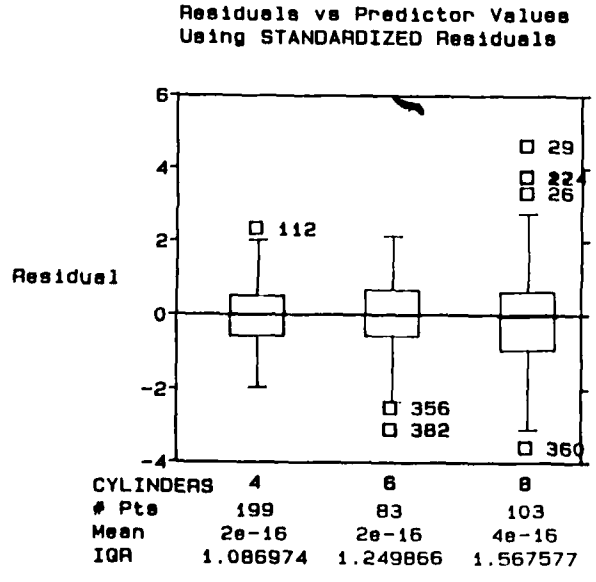


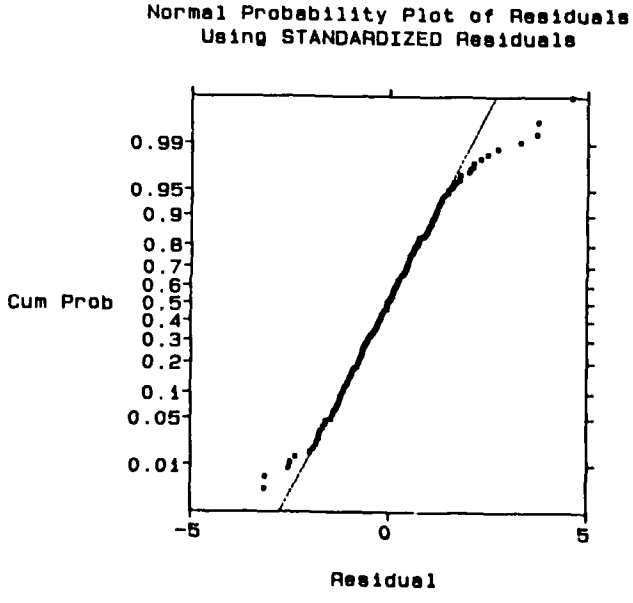
Residuals vs Fitted Values of GPM100
Using ABSOLUTE STUDENTIZED Residuals



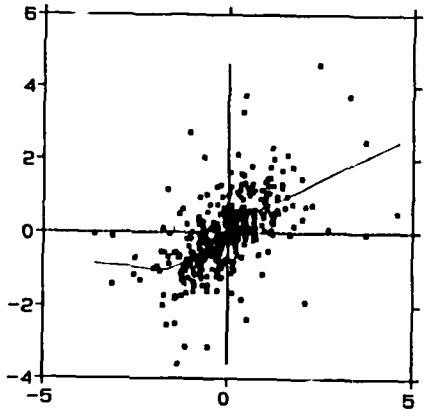

Residuals vs Predictor Values
Using STANDARDIZED Residuals
Residual
WT100
CYLINDERS-4
CYLINDERS-6
CYLINDERS-8


Residuals vs YEAR Values
Using STANDARDIZED Residuals
Residual
YEAR

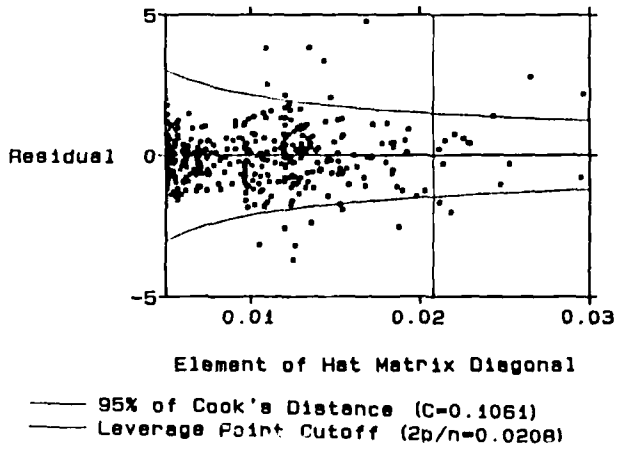

Residuals vs Predictor Values
Using STANDARDIZED Residuals
Residual
112
356
382
29
224
26
360
CYLINDERS 4 6 8
Pts 199 83 103
Mean 2e-16 2e-16 4e-16
IQR 1.086974 1.249866 1.567577


Normal Probability Plot of Residuals
Using STANDARDIZED Residuals
Cum Prob
Residual

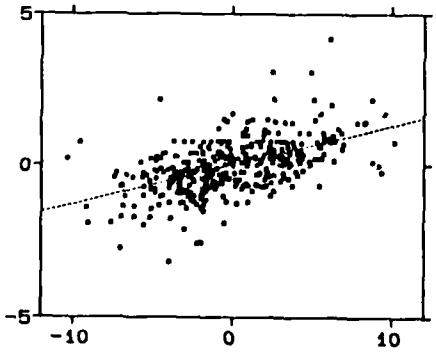
Lag-1 Serial Graph of Residuals
Using STANDARDIZED Residuals



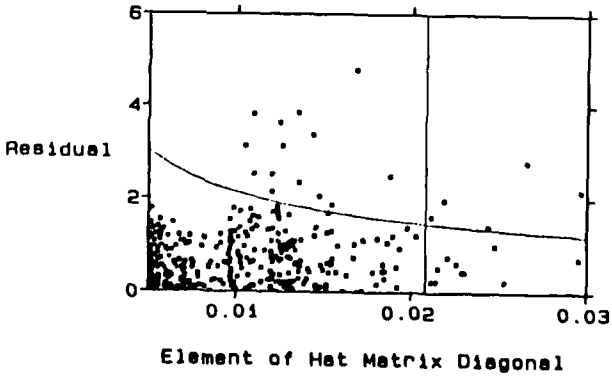
Influence Plot of Residuals of GPM100
Using STUDENTIZED Residuals



Partial Regression Plot of GPM100
for WT100



Influence Plot of Residuals of GPM100
Using ABSOLUTE STUDENTIZED Residuals



4. SUMMARY

The RS/Explore computing environment

- integrates numerical and graphical summaries
- provides convenient storage of results
- maintains a diary of data analysis

The RS/Explore menu system

- organizes model-building process
- highlights appropriate steps
- requires no programming effort

The statistical tools for residual analysis in RS/Explore

- allow a variety of scaled residuals
- provide lowess smooth on all scatterplots
- give easy access to regression diagnostics

REFERENCES

BBN Software Products Corporation (1987). *RS/Explore User's Guide*. BBN Software Products Corporation, Cambridge, MA.

Belsley, D. A., E. Kuh, and R. E. Welsch (1980). *Regression Diagnostics*. John Wiley, New York.

Cook, R. D. (1977). 'Detection of Influential Observation in Linear Regression'. *Technometrics*, **19**, 15–18.

Cook, R. D. (1979). 'Influential Observations in Linear Regression'. *Journal of the American Statistical Association*, **74**, 169–174.

Draper, N. R., and H. Smith (1981). *Applied Regression Analysis*, second edition. John Wiley, New York.

AN EXPERT SYSTEM FOR COMPUTER-GUIDED SIGNAL PROCESSING AND DATA ANALYSIS

David A. Whitney
Ilya Schiller

The Analytic Sciences Corporation
55 Walkers Brook Drive
Reading, MA 01867

ABSTRACT

This paper describes the application of expert system technology to the development of a software tool for processing and analysis of time series signals. The system integrates distinct numerical and symbolic processing cores to form an analysis environment where numeric and symbolic processing tasks are performed as needed during the analysis. Sophisticated off-the-shelf numerical analysis software is coupled with a high-end expert system development shell to form the integrated system. A knowledge base of rules for performing ARIMA (AutoRegressive Integrated Moving Average) time series modeling is implemented in the system prototype. The user interface is presented in a multi-window environment on a workstation with bit mapped graphics.

1. INTRODUCTION

This paper describes a prototype expert system for signal processing and data analysis, called COSTAR (COordinated Statistical Analysis and Reasoning). As illustrated in Fig. 1, the philosophy behind the system design is to integrate four key functional components of the data analysis process: Users, Graphics, Symbolic Rules, and Numerics. The system can be used as a "black box", but it allows intervention of the user at selected points in the analysis. Its objective is to serve as an example of how these functional areas can be integrated for complex signal analysis, as well as to serve as a test bed for studying issues of strategy development and rule refinement. General references for current work in the area of artificial intelligence and expert systems in statistics are Gale (1986) and Haux (1986). Discussions of other specific systems can be found there and in Gale and Pregibon (1984), Milios (1984), Nelder (1986), and Nii and Feigenbaum (1978).

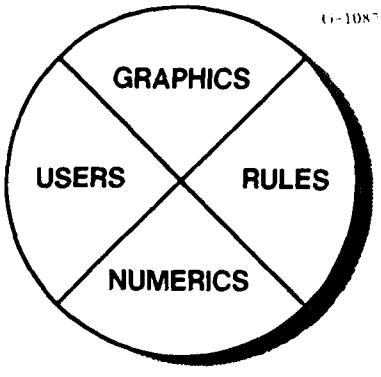


Figure 1 System Functional Integration

2. STATISTICAL PROBLEM ADDRESSED

One of the objectives of this work is to study and formalize analysis strategies for statistical signal processing and modeling. This was also one motivation in the development of pioneering systems such as DINDE (Oldford and Peters, 1988) and DART (Donoho, 1984). In the development of COSTAR, inferences about general statistical strategies for signal analysis and modeling are developed by studying a particular modeling problem with widespread applications: ARIMA modeling for a class of nonstationary univariate time series. These techniques begin (after outlier identification, editing, detrending, and variance stabilizing transformations) with linear differencing operations to transform a nonstationary series to a stationary one. This is followed by an analysis of the autocorrelation (ACF) and partial autocorrelation (PACF) functions to identify patterns that are characteristic of different model structures and orders. Model parameters are then fit to the series, using an iterative maximum likelihood estimation scheme in the general ARIMA case. This is followed by residual analysis to score or rank the goodness of the fitted model for comparison with other candidate models. For detailed descriptions of the ARIMA modeling techniques applied in this work, see Box and Jenkins (1976) or Brockwell and Davis (1987). Section 6 of this paper presents an example showing several of these modeling stages.

While the emphasis here is on ARIMA model *fitting* as the objective of the signal analysis, such models are widely used for forecasting, and can also be combined with intervention analysis to detect discrete changes in parameters of a model caused by exogenous events.

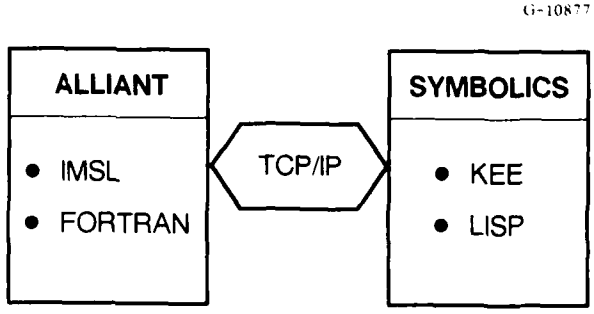


Figure 2 Hardware and Software Integration

3. SYSTEM ARCHITECTURE

Effective automated signal processing and data analysis requires the integration of *symbolic* and *numeric* processing. In this system, the symbolic processing burden is carried by the rule based expert system and, in a limited and structured way, by the user. The numeric processing is handled by a modern "number crunching" system. A schematic of the system architecture is shown in Fig. 2. A mini-supercomputer, the Alliant FX/8 parallel processor running under Concentrix 3.0, performs the numeric processing using a collection of 15-20 routines from the Fortran based IMSL 9.2 library (IMSL, 1984). The computational requirements for univariate time series modeling do not really demand a supercomputer, but this architecture will make it easier to address more computationally intensive signal analysis within this system framework in the future. Symbolic processing is handled by a Symbolics 3640 running under Genera 7.1 and using the expert system development shell KEE (Knowledge Engineering Environment) 3.1 (Intellicorp, 1987) and Common Lisp. Information is exchanged between numeric and symbolic processing entities over a network operating under the protocols of TCP/IP (Transmission Control Protocol/Interface Protocol). The user interface is provided

through the Symbolics workstation, which displays bit-mapped graphics in a multi window environment. User inputs are provided through a keyboard or a 3-button mouse.

The system reflects an integration of distinct hardware and software functions. The functional outputs are integrated, but the numeric and symbolic elements remain distinct, passing information across the network to each other. Advantages of such a configuration are that each type of *hardware and software that is most appropriate for an aspect of the task is applied*, and existing numerical signal and data analysis software can be used directly. A *disadvantage is the need for the developer to work in and integrate* two different hardware and software environments – this has not proved to be a major problem, however. While this (Fig. 2) is a "high-end" hardware configuration, it is now possible to build a more economical system with similar functionality using KEE and IMSL on a 80386 PC system running under a version of UNIX.

4. THE KNOWLEDGE BASE

The knowledge base (KB) uses structures that are becoming common in sophisticated expert system software, due to their generality and power. These are frames and object-oriented programming techniques. The knowledge base is composed of three major segments: *data objects*, *production rules*, and *graphical objects*. The KB structure is defined so as to reflect the natural structure of the modeling problem addressed by the analysis.

The basic objects in the system are data sets existing at a particular stage of the analysis. For example, the knowledge base may initially contain a single "raw" data set object. The operation of data editing will create an edited data set, the operation of transforming, a transformed data set. If alternative transforms are entertained, for example, multiple transformed data sets corresponding to a single edited data set may be part of the knowledge base. Figure 3 shows a fragment of a hierarchical tree showing some of the types of objects in the knowledge base (Figs. 3-8 are displayed at the end of this article).

Data objects are represented by frames, which hold not only the time series of data, but a variety of additional information about a particular kind of data object. For example, a data object in the residual class has a frame with slots that hold information about the summary statistics of the residuals, the model used to derive the residuals, scores for goodness of fit tests, etc. Figure 4 shows a portion of the residual frame attributes for this system's KB. Seven classes of data objects are defined in the current system. Data objects are defined as members of *prototype* classes and inherit attributes appropriate to that particular class. The notion of prototype was defined in Aikins (1983) and has also been applied in the REX system for regression analysis (Gale 1986a). In addition to helping formalize and streamline the structure of the expert system KB, the idea is useful in strategy formalization, *since it directs the statistician to think more abstractly about:*

- The types of objects that form the analysis task
- When they are needed
- How they generalize to or specialize from other classes of objects

The primary repository of signal analysis expertise is contained in the *production rule system* segment of the knowledge base. This is a set of *if-then* rules that perform the symbolic reasoning in the system, as well as controlling the numeric processing. Approximately 50 rules are implemented in the first generation COSTAR system, but the knowledge base is undergoing revision and the number of rules in the next generation system is expected to at least double. The rules are generally invoked in a forward chaining process, consistent with the "data-driven" nature of a complex signal analysis. The rules in the KB are partitioned into rule classes that correspond approximately to the way that data objects are divided into classes. For example, a class of "residual rules" that are appropriate only for application to residual data objects is defined. Rules are defined in a hierarchy, as shown in Fig. 5. Rule classes are divided into subclasses, and generic subclasses contain specific instances of particular rules (e.g., the Box Ljung Scoring Rule in the Residual.Rules Class in the ARIMA Modeling Rules Super Class shown in Fig. 5).

This hierarchical, categorical structuring of rules in the knowledge base serves several purposes. It allows for structured development of the KB – portions of the overall analysis can be developed independently. It also permits a more organized debugging of the knowledge base at an intermediate level. Instead of evaluating each rule individually, or evaluating the final system conclusions based on firing all the rules, reasoning can be invoked using a particular rule subclass. This allows for:

- Analysis of the integrated performance of a subset of rules
- Scaling down the complexity of problem diagnosis significantly.

In addition to the *logical* hierarchy reflected in Fig. 5, the rules in the KB also reflect a *conceptual* hierarchy that is derived from an idea of the generic signal analysis process. The rules can be labeled in one of three ways:

- Strategic
- Tactical
- Mechanical

These labels reflect the ways in which knowledge seems to be provided by experts during the "knowledge engineering", i.e., expertise extraction, phase of KB development. For example, a *strategic* rule may describe at what phase of the analysis checks for non stationarity are appropriate. An associated *tactical* rule could identify ways to test for stationarity, such as examination of patterns in ACFs and PACFs. *Mechanical* rules would have heuristics for identifying patterns in computed correlation functions, as well as triggers for invoking those numerical procedures needed to compute correlations if they had not already been computed. Such a *conceptual rule hierarchy* is clearly useful for identifying and formalizing statistical analysis strategies.

The knowledge base also contains *graphical objects* and rules for controlling them. Rules for the creation and display of different types of plots, tabular summaries, and icons or pushbuttons, are *contained in* the knowledge base as well. The object-oriented structure of the system allows procedures for generating and displaying graphic objects to be linked directly to the data objects themselves as frame attributes. This helps to:

- Organize the KB
- Remind the knowledge engineer that graphical displays play a role on a par with numeric data in signal analysis problems.

5. SYSTEM CONTROL STRATEGIES

Many of the difficult problems in developing an effective software tool for computer guided data analysis seem to arise in the area of *system control*. This is, after all, the area in which a data analyst shows most of his reasoning expertise: what numerical computation should be performed at what stage, and what should be done next as a result of the numerical computations performed so far? While expertise in interpreting the results of a particular numerical procedure is often needed, *the need for expert control of the overall analysis is dominant*. Control strategies in this software system are currently being refined. This section outlines the planned control strategies.

As mentioned in the introduction, the user plays a role in controlling the system through a limited number of interaction opportunities presented by the system. These inputs take the form of either information supplied by the user at initial prompts, or by mouse indicated confirmation of veto of options selected by the system at various stages of the analysis. For example, Fig. 6 shows a USER OPTIONS icon in the CURRENT ACTIVITY window with four types of variance stabilizing transforms that are available in the system. The transform highlighted in black indicates the system's recommendation for a transformation, based on the rules in

the KB. The user may, however, use the mouse to highlight another choice of transform and override the system recommendation. The system control structure invokes a break in the forward chaining agenda at times that are appropriate for such a user selection.

The system must have strategies and control structures for generating, maintaining and ranking multiple alternative models for the time series data. At each stage in the analysis (where a stage is defined as the generation of a particular instance of a prototypical object, e.g., a specific set of edited data), one or more alternative objects (parts of a candidate modeling hypothesis) are generated. These objects are scored according to a "certainty" in the facts or rule conclusions that led to their creation. For example, an edited data set created by deleting a point that has magnitude larger than five times the interquartile range of the raw sample may have a very high certainty score. An edited data set created by deleting that point, plus another point with magnitude exceeding twice the interquartile range, may be excluding a marginal outlier, and so has a lower certainty score. The ranked candidates are placed on a stack, and the item on the top of the stack is removed and used for the next stage of the analysis. This stack is constructed as progress is made through each stage of the analysis.

When a single candidate model has been fitted and scored, other candidate models at that stage are also fitted and ranked. Depending on the quality of the fitted models, a loop-back to an earlier stage of analysis may be required. For example, if candidate ARIMA (p,1,0) models for several choices of AR order p do not produce residuals with the desired whiteness properties, then a loop back to a model structure selection phase may be needed to consider ARIMA (p,1,q) models. If fits still seem inadequate, a deeper loop back to a model differencing order stage may be required, where the stack containing candidate differencing orders is used to obtain the next candidate differencing order. Forward chaining toward model structure selection and fitting is then resumed.

Several important issues in the above scheme need to be addressed. The first is in the use of the term "certainty". This is not to be taken in a formal probabilistic sense, or even necessarily in the sense of a belief function formalization (Kanal et al. 1986). Here we mean only some *heuristic scoring* which ranks choices at a particular stage, either by strength of a numerical result (e.g., observed significance level) or lack of other viable alternatives. Schemes for combining these uncertainties across stages may require inclusion of a more elaborate calculus of uncertainty, however. This is particularly true when multiple candidate models, derived from multiple loop backs of varying depths, need to be finally ranked for presentation to the user.

Another issue is the determination of the depth of a loop back from a particular stage. The simplest heuristic requires a loop-back of one level to the previous stage, where locally-ranked alternatives are then sorted. However, rankings of alternatives at an earlier stage may now need to be updated, *conditioned* on what had been observed at the later stages. For example, if residuals from fitting a collection of ARIMA (p,1,0) models show residuals with distinct changes in mean level over the series (a remaining nonstationarity), an expert may rank an alternative ARIMA (p,1,q) structure very low, and wish to loop back two levels to consider ARIMA (p,2,0) structures. By using a second order differencing operator, instead of the original first order difference, the expert hopes to remove the observed residual nonstationarity. This may call for the development of much more elaborate schemes for handling uncertainty if true expert analysis strategies are to be studied and emulated. This is an area of ongoing research.

6. AN EXAMPLE OF THE SYSTEM

Figures 6 through 8 show displays for several of the latter stages of a typical signal analysis session. The aim is to show what the user interface looks like and what types of data are displayed. Figure 6 shows a stage where edited data are being examined to determine if a variance-stabilizing transform is required. The edited data (from an actual ARIMA (2,1,0) model) are displayed along with a table of summary statistics. A possible first-order nonstationarity in the data can be seen in the plot. The CURRENT ACTIVITY window indicates that no variance-stabilizing transform is recommended. An icon appears in the left portion of the window and presents options which the user can select by clicking on the left button of the three button mouse. Two other windows are visible in the display. The LISP LISTENER window provides a deeper level of access to the system than the average user will employ. The ANALYSIS SCRIPT window provides a *trace* of the steps in the analysis that have been performed so far. Such a trace will be used in the future for *formalized rule refinement* based on deductions from application-specific data analysis sessions.

Figure 7 shows an analysis of the transformed data to check for nonstationarities. A simple set of pattern recognition rules "examine" the ACF and PACF, classify the correlation patterns into one of five characteristic classes, and display the results in mouse-sensitive icons. Plots of the ACF and PACF are also displayed for the user to view and analyze.

Figure 8 shows a display of the full COSTAR system screen after a model has been fit and evaluated. The CURRENT ACTIVITY window has five new icons which display information about the fitted ARIMA (1,1,1) model (an incorrect model structure was fit for illustration). These icons are for display only, and are not subject to mouse control. These icons display the fitted model parameters, a composite model goodness score (POOR), and the (normalized) numerical results of a Box-Ljung diagnostic test on the residuals. Another plot has been displayed in the INTEGRATED PSD window, the integrated power spectral density of the residuals. The linearity of this curve reflects the degree of whiteness of the residuals. Various windows and displays appear and disappear as appropriate in fixed locations on the screen as the analysis progresses. This is aimed at avoiding the clutter and confusion that can result from *too much* flexibility in window generation and rearrangement.

7. SUMMARY AND FUTURE WORK

This paper has described the implementation of COSTAR, a prototype expert system signal analysis and data processing tool now under development at TASC. Descriptions of the system architecture and knowledge base, and examples of the operation of the current system have been given. Current work focuses on enhancement of the system knowledge base and elaboration of the modeling process control structure to handle the complexities of multiple, multi-stage loop backs. Future work will focus on the use of analysis traces for automated rule-refinement in fielded systems, the application and development of more formalized validation procedures for production rule systems, and the "self-validation" of a system - e.g., procedures that allow the system to tell when it may be faced with a cyclostationary process, a nonstationarity process which cannot be adequately described by the available model structures.

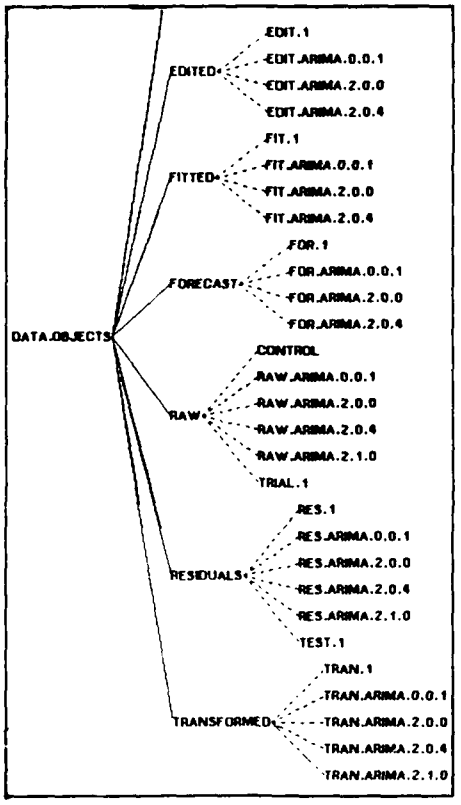


Figure 3 Data Object Hierarchy

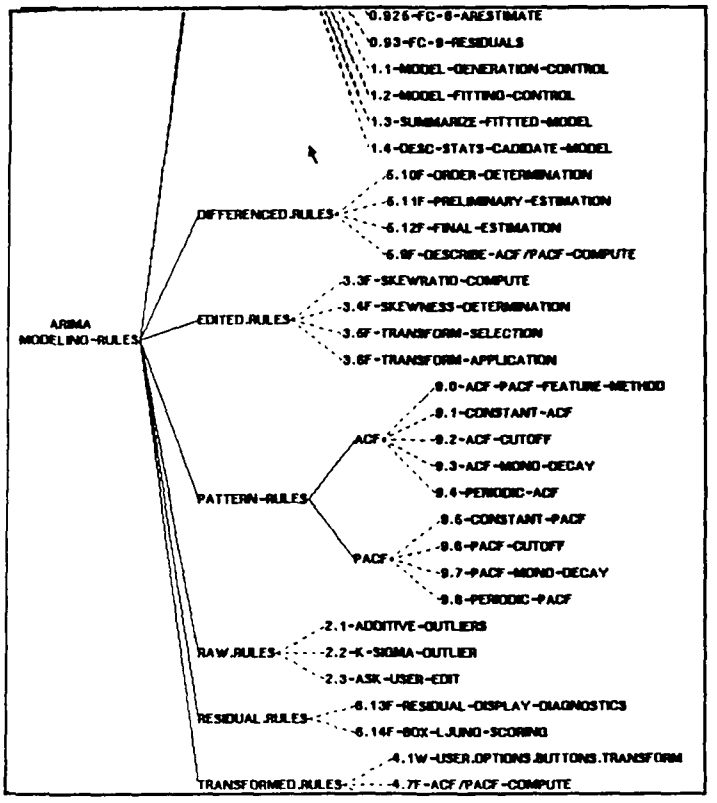


Figure 5 Analysis Rule Hierarchy

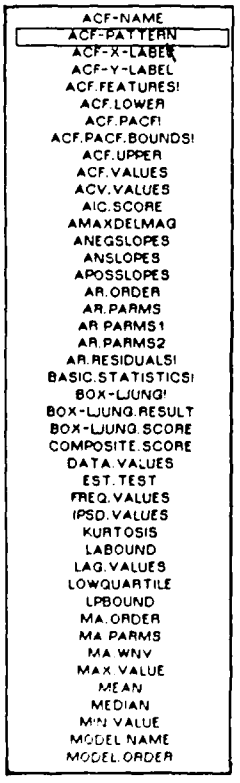


Figure 4 Data Object Frame Attributes

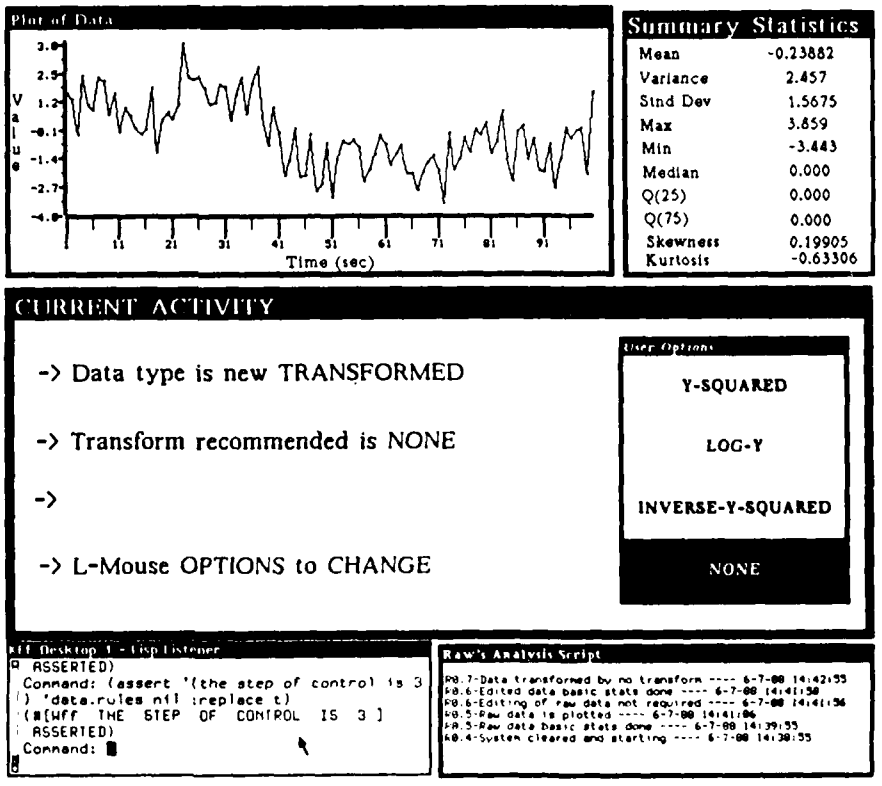


Figure 6 Stationarity Transformation Recommendation

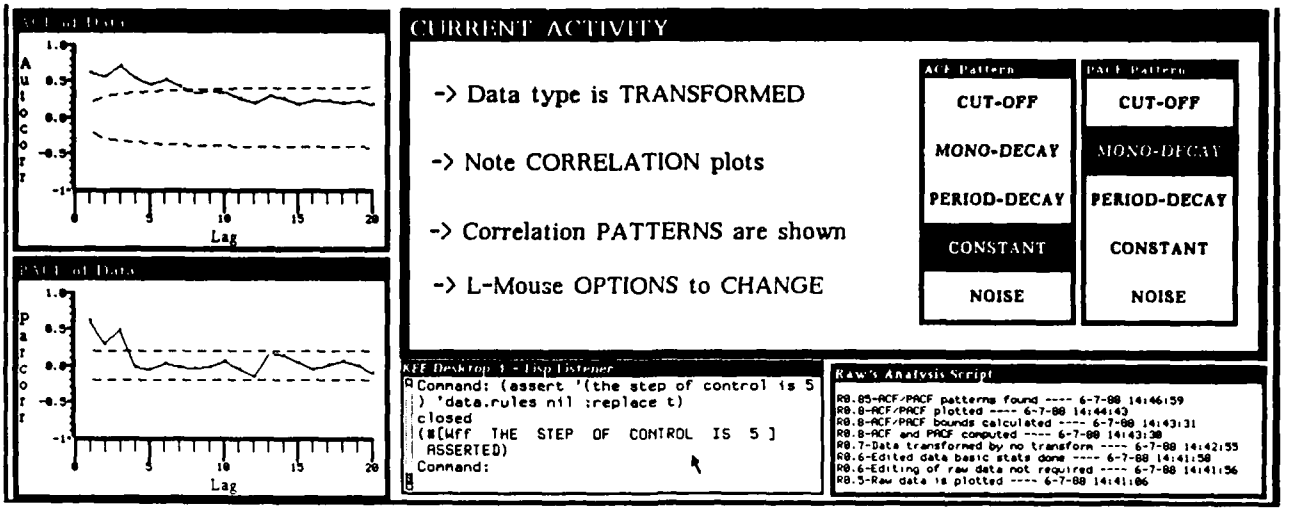


Figure 7 Correlation Pattern Identification

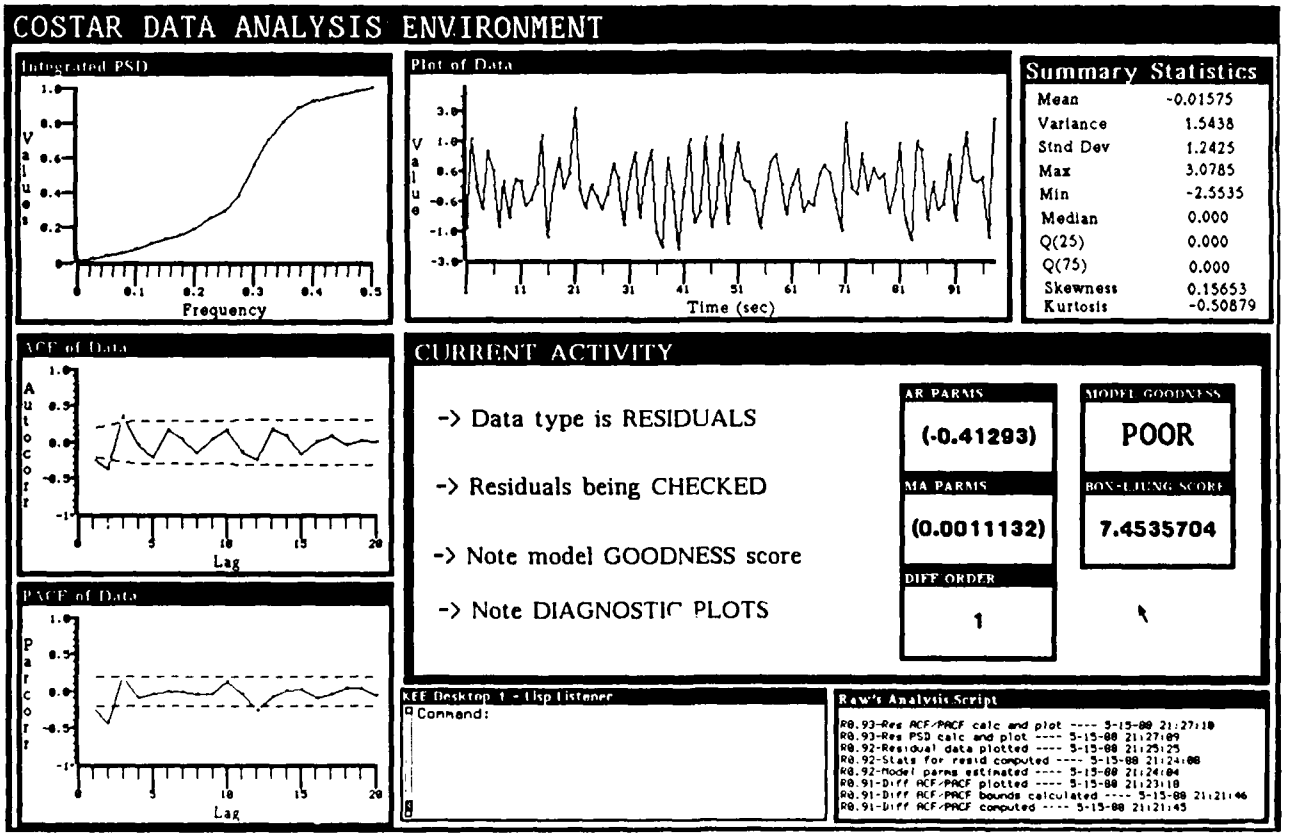


Figure 8 Residual Analysis and Model Evaluation

REFERENCES

1. Aikens, J.S. (1983). "Prototypical Knowledge for Expert Systems", *Artificial Intelligence*, No. 120, pp. 163-210.

2. Box, G.E.P., and Jenkins, G.M. (1976). *Time Series Analysis, Forecasting, and Control*, Holden-Day, Oakland, CA.

3. Brockwell, P.J., and Davis, R.A. (1987). *Time Series: Theory and Methods*, Springer-Verlag, New York, NY.

4. Donoho, D.L. (1984). *DART: A Tool for Research in Data Analysis*, Unpublished Ph.D. Thesis, Harvard University, Cambridge, MA.

5. Gale, W.A., and Pregibon, D. (1984). "REX: An Expert System for Regression Analysis", *COMPSTAT 1984: Proceedings in Computational Statistics*, Physica-Verlag, Vienna, Austria, pp. 242-248.

6. Gale, W.A. (1986), ed., *Artificial Intelligence and Statistics*, Addison Wesley, Reading, MA.

7. Gale, W.A. (1986a). *REX Review*, in Gale (1986).

8. Haux, R. (1986), ed., *Expert Systems in Statistics*, Gustav Fischer, New York, NY.

9. IMSL (1984). *IMSL Library User's Manual Edition 9.2*, IMSL, Houston, TX.

10. Intellicorp (1987). *KEE Software Development System User's Manual Version 3.1*, Intellicorp, CA.

11. Kanal, L.N., and Lemmer, J.F. (1986). *Uncertainty in Artificial Intelligence*, North Holland, New York, NY.

12. Nelder, J.A., and Wolstenholme, D. (1986). "A Front-End for GLIM", *Proceedings of the 18th Symposium on the Interface*, ASA, Washington, D.C.

13. Nii, H.P., and Feigenbaum, E.A. (1978). "Rule Based Understanding of Signals", in *Pattern Directed Inference Systems*, Waterman and Hayes-Roth, eds., Academic Press, New York, NY.

14. Oldford, R.W., and Peters, S.C. (1988). "DINDE: Towards More Sophisticated Software Environments for Statistics", *SIAM J. Sci. Stat. Comput.*, Vol. 9, No. 1, pp. 191-211.

IX. ARTIFICIAL INTELLIGENCE, EXPERT SYSTEMS, AND STATISTICS

P_ITSS_A — A Time Series Analysis System Embedded in Lisp

Donald B. Percival and R. Keith Kerr
Applied Physics Laboratory, HN-10,
University of Washington
Seattle, Washington 98105-6698

We describe the design of P_ITSS_A, a computer system for interactive time series and spectral analysis. This system is written in Lisp, a language which has long been a favorite of researchers in computer science but which has not been used extensively for data analysis. Some of its interesting features include the use of object-oriented programming to break up P_ITSS_A into a large number of separate modules; a systematic way of defining interactions between the user and P_ITSS_A via a graphical input device ("mouse"); and an implementation of a number of enhancements to a standard time series package which leads to a qualitative improvement in interactive data analysis for time series.

1. INTRODUCTION

We describe in this paper P_ITSS_A— a computer system which supports interactive time series and spectral analysis and which is written in the language Lisp. Our system is part of an effort by a small number of investigators in recent years to implement some of the ideas put forth in a series of articles by McDonald and Pedersen (1985a, 1985b, 1988). The basic thesis of their work is that interactive data analysis is best supported by hardware and software originally designed by computer scientists to efficiently support experimental programming. These include, on the hardware side, a modern computer workstation with high-speed and high-resolution bitmap graphics, and, on the software side, a computing environment such as is provided by a modern programming environment for, say, the language Lisp.

There are several research questions we are currently investigating with P_ITSS_A, among which we will be concerned with three in this paper. First, how can data analysts best take advantage of the opportunities afforded by the hardware and software supplied on modern computer workstations? Most (but by no means all) of the widely used software systems for interactive data analysis are packages which were originally designed on a batch-processing system. These typically make limited use of the capabilities of modern workstations (other than superficial use of menus to replace typing of certain commands). A few systems (such as S (Becker and Chambers 1984)) were originally developed in an interactive computing environment (such as UNIX). These are a vast improvement over batch-oriented systems, but it is necessary to be somewhat of a computer expert to augment them to handle graphical interaction with a user.

Second, is is possible to design an interactive data analysis system which is accessible by, and useful for, users of many different levels of sophistication? Typically, "user friendly" systems can drive an expert user to distraction with their bulky use of menus, while terse systems for an expert are incomprehensible to novices. Two things are desirable here: a system with a good support for novices but which can easily be "opened up" for fundamental modifications by an expert; and a system which grows in usefulness as novices learn more and more about both it and interactive data analysis and become experts themselves.

Third, what new forms of interactive time series and spectral analysis are possible with modern workstations? Time series analysis is a field which has been particularly influenced by available computing power. Much of the emphasis on lag window (or Blackman-Tukey) spectral estimators in the 1950's was due to computational issues: the lag window approach allowed spectral estimates to be calculated from only a small number of sample autocovariance function values. With the advent of the Fast Fourier transform and more powerful computers in the 1960's, it became possible to use spectral estimation techniques with a greater computational overhead. The computationally intensive multiple taper approach to spectral analysis (Thomson 1982) would have been a theoretical curiosity if it had been introduced 15 years ago. The advent of computer workstations opens up new avenues for qualitatively improving interactive time series analysis.

P_ITSS_A is a continually evolving experiment which seeks to address these (and other) issues. Our first working version was developed in 1984 on a Symbolics Lisp Machine, one of the few workstations at that time which could support the type of inter-

active system we were interested in designing. The rapid growth and development of computer hardware in the past 4 years now makes it feasible to make our work more widely available. For example, we are now in the process of porting a portion of P_ITSS_A to an Apple Macintosh II, which is more than an order of magnitude less expensive than a Lisp Machine was four years ago.

P_ITSS_A has been used in several graduate level classes in time series and spectral analysis over the past four years, both for in-class demonstrations of data analysis techniques and for use by students in class projects. It has also been demonstrated to dozens of colleagues and groups of visitors. Discussions with individuals who have either seen P_ITSS_A demonstrated or used it extensively have lead to a number of fundamental changes in the underlying design of P_ITSS_A. While we are fairly happy with its form as reported in this article, we plan to continue to use it as a test bed for new ideas in the future.

2. WHY LISP?

The question posed in the heading is the one most frequently asked by colleagues who have seen demonstrations of P_ITSS_A. Our rationale for embedding a statistical system within Lisp is discussed in the subsections below. We remark here that there are languages other than Lisp which have some (or all) of its desirable features and would be a reasonable alternative choice (Smalltalk is a prime example). Lisp does enjoy considerable popularity in the computer science community. It is available and supported efficiently on a number of different computers (ranging from special workstations designed specifically to support Lisp — the so-called Lisp Machines — to personal computers). This advantage is offset somewhat by a profusion of different dialects of Lisp — a problem which stimulated the recent definition of Common Lisp as a proposed standard (Steele 1984).

2.1 Interpretation and Compilation

The simplest systems for interactive data analysis work in the following way. The user types in a command; a command processor (supplied by the system designer) interprets the command and does something — returns one or more computed values, assigns a value to a variable, or displays a plot; and the user looks at the results and types in a new command. The designers of such systems supply the user with a certain number of basic commands with which to work. Although this number is quite large for sophisticated systems, no such system is ever complete. It is impossible for designers to anticipate the needs of users (particularly since interactive data analysis is most often exploratory in nature). The question then arises as to how to let the user extend the statistical system to met his or her needs.

A certain amount of flexibility is introduced by allowing the user to write *macros*. Macros greatly decrease the amount of typing which a user must do by packaging together commands into groups — a single macro command expands into many basic commands. However, macros have two problems. First, the user is really using the command language of the statistical system as a programming language. This means that the design and complexity of the tasks which can be accomplished by macros is usually limited because the command language was designed to convey statistical instructions to the computer and not to be a general purpose programming language (with support for loops, conditional execution, block structure, complicated data structures, and so forth). This can be remedied, of course, if the system designer is willing to take the time to augment the macro facility to include many of these programming features.

A second problem with macros is that, after a user invokes a macro, the commands of which it is composed are interpreted and executed by the command processor one at a time. If the macro expands into a hundred commands, the command processor must interpret and execute each of these commands. The overhead of command processors is usually not negligible. This means that execution of macros can be quite slow. Again, there is solution to this problem, but it means that the system designer must provide for compilation of macros into efficient machine code instead of just interpretation of their contents.

More sophisticated statistical systems (such as S (Becker and Chambers (1985))) provide a second (and more powerful) way for a user to deal with problems which cannot be handled by the basic commands. Here the user is allowed to augment the set of basic commands by defining new ones. The code which defines these new commands in written in a full-fledged programming language (such as Fortran or C), and the code for these commands is compiled (i.e., translated into efficient machine code) to allow rapid execution. Once defined, these new commands enjoy the same status as the basic commands supplied by the system designers.

There are two problems with this way. First, ideas for new commands are often inspired by the results of interactive data analysis within the statisti-

cal system. That means that the new command has in effect already been implemented once in the form of groups of commands and macros which have been pieced together. The user must begin over again because the command language of the statistical system and the programming language which is used to define commands are different.

The second problem is the lack of extensive support for debugging within a statistical system. The designers of such systems usually assume that code for commands has been thoroughly debugged. Unfortunately, subtle bugs often occur in supposedly well-tested routines. If this occurs in the code for one of the commands in a statistical system, the user usually has to revert to writing a driver program in the native language which is used to define the commands and do the debugging entirely outside of the statistical system. This process can be quite time consuming.

How does use of Lisp improve this situation? Lisp is an interpreted language. The user types a Lisp expression, and the command processor for Lisp (known as the "reader") interprets it and returns a value. The value the reader returns can be as simple as a single number or as complex as an entire function. By this means, the user can evoke various defined functions and set (or, in Lisp terminology, bind) the values of these functions to various symbols for later reference. It is rather trivial to develop the basic structure of an interactive statistical system in Lisp. The Lisp reader plays the role of the command processor, and Lisp functions play the role of basic commands. The equivalent of macros within Lisp is simply other Lisp functions, since any Lisp function can make use of any other Lisp function. The real task is thus to design a set of functions useful for interactive data analysis. From the viewpoint of a potential user, learning enough Lisp to make use of this system is no more difficult or time consuming than learning to use a command-driven system such as S or MATLAB.

A number of benefits immediately follow. First, since Lisp is both a interpreted language and a compiled language, Lisp statistical "macros" can be executed as efficiently as any other function in Lisp. Second, because Lisp is a full fledged programming language, it has an extensive range of features for loops, conditionals, and handling quite complex data structures. The user does not have to try to program in a command language designed primarily to facilitate interactive data analysis. Third, because a statistical system written in Lisp uses that language both as a command language and a programming language, ideas for new data analysis functions which arise in the course of an interactive data analysis as small test functions can be readily repackaged as new Lisp functions. Fourth, because Lisp is used so extensively in the computer science community, extensive debuggers have been built for it. These allow the user to quickly track down bugs in his or her program (or to detect errors in existing Lisp functions) — all without having to ever leave the Lisp system.

2.2 Complex Data Structures

When the user gives a command to the command processor of an ordinary statistical system, the command processor returns a value. In simple systems, this value may be a single number or an array of numbers; in more complex systems, it may be a data structure, each slot of which contains a number or an array. Command processors rarely deal with more complex data structures than these. In contrast, the Lisp reader can return values which are considerably more complex, allow greater flexibility, and correspond more closely to the way in which a statistician thinks about a problem. The following simple example illustrates these ideas.

Suppose that we have a Lisp function which fits an autoregressive (AR) model to a time series. Now a fitted AR model can be used to estimate the spectral density function (sdf) of a series. What is the best way to represent this estimated function? The usual approach is to express it as a vector of values computed over a grid of equally spaced frequencies.

In Lisp, however, we have the option of representing it actually as a function — the result of executing a Lisp function can be to return a new function. This means that we can treat the estimated sdf as a true function — it can be numerically integrated and differentiated, and its peak values can be searched for to within any precision desired. This later capability is particularly important, since a common error in displaying an AR sdf is the failure to properly evaluate it around sharp peaks (Burg 1975). Representation of the sdf as an actual function makes its easier to design code to do this.

2.3 Different Language Paradigms

The predominant language paradigm in use by data analysts is called procedure-oriented programming. This is the style of programming supported by such languages as Fortran and C, where the basic module is a subroutine or function. Lisp can support this style of programming, but it has also proven flexible enough to support many other different paradigms proposed in computer science over the

years. Among these are object-oriented programming (discussed in greater detail in the next section), constraint-oriented programming, and access-oriented programming. Each of these paradigms is useful in certain problem areas and, in particular, is of potential use to support interactive data analysis (see, in particular, McDonald (1986)).

3. WHY OOP?

Object-oriented programming (OOP) has been the subject of considerable attention in recent years in the computer science community. Its use in the statistical community has been fairly limited (for examples, see Stuetzle (1987), McDonald (1986), and Oldford and Peters (1986)), but we feel that it offers a number of advantages for constructing a statistical systems over procedure-oriented programming. The specific features of OOP which have facilitated our development of P_ITSS_A are discussed in detail in the subsections below, but first we make a few subjective comments.

OOP is somewhat like structured programming in that it gives systems designers a specific approach for organizing, maintaining, modifying, and extending large programs. For some problems, it seems a more natural way to approach the programming problem than structured programming, because it allows program design to follow rather closely the way a program appears from a user's point of view (as operations on various objects — entities with a separate identity within a computer). This yields a number of benefits. First, it allows a user more easily to develop a mental model of how a program will react to certain actions which her or she takes. Second, it allows a system designer to go in and make changes to existing code more quickly — if the code matches the way a program is perceived to work, it is easier to know where to make changes.

While OOP has enjoyed considerable success for programming problems where there is a more or less natural decomposition of the problem into objects (such as in the simulation of a paper mill, where the objects correspond to physical entities such as paper rollers), it is less obvious how it can be used to construct a statistical system. We hope that the reader is convinced of its usefulness by the end of this paper. We can report, however, that we are now on our third major redesign of P_ITSS_A in a four year period and that the claims of advocates of objected-oriented programming as to its benefits in terms of modifiability and maintainability are true. In fact, each of the redesigns has lead to the definition of more objects and a *stronger* use of the language paradigm. However, our experience also shows that there are subtlies in OOP that are not apparent to the novice (at least to novices who were initially trained in the more traditional procedure-oriented programming).

3.1 Classes

A key concept in OOP is that of a class of objects. All objects in a particular class share a particular data structure. For example, one class in P_ITSS_A is called **ordered-x-y-pairs**. Every object in this class has three *slots* (sometimes called *instance variables*). The symbolic names for these slots are *ordered-x-values*, *y-values*, and *number-of-pairs*. Typically *ordered-x-values* and *y-values* are (pointers to) vectors of length *number-of-pairs*, and *ordered-x-values* is assumed to have its values ordered. A real-valued irregularly sampled time series of length 100 could be (at least partially) represented as a particular object of this class. We would need to bind (assign) the slot *ordered-x-values* to a vector of length 100 with the times at which the time series was sampled; the slot *y-values* to a similar vector with the values of the time series at each of the 100 times; and *number-of-pairs* to the value 100. A second object of this class (used to represent, say, a second time series) typically would have different bindings (assignments) for one or more its slots.

3.2 Generic Functions (Message Passing)

A second fundamental notion in OOP is that of a *generic function*. We illustrate the idea behind this concept with an example from P_ITSS_A. Two of its classes are called **real-time-series** and **complex-time-series**, which are used to represent real-valued and complex-valued time series, respectively (how they are related to the class **ordered-x-y-pairs** is discussed in the next subsection). A popular way of fitting an autoregressive model to a time series is by means of Burg's algorithm (Marple 1987). There are two different univariate versions of this algorithm, one for real-valued time series, and one for complex-valued series. Suppose that we create a generic function called "burg" which takes as input an object of either the class **real-time-series** or **complex-time-series**. If "burg" is a generic function, its definition depends upon the class of its input. Thus, if we apply "burg" to an object belonging to the class **real-time-series** (**complex-time-series**), its definition would be a routine which implements Burg's algorithm for real-valued (complex-valued) series.

An equivalent way of expressing this idea is as *message passing*. Here we conceptually send a message to an object, and the object responds in a particular way — the response depends on what class

it belongs to. Thus, if we send the message "burg" to an object belonging to the class **real-time-series** (**complex-time-series**), the object responds by applying Burg's algorithm to the real-valued (complex-valued) time series which it represents.

There are two distinct advantages to this approach. First, we can reduce the number of commands which we need to know. We need only remember that "burg" is the proper message to pass to (or generic function to use with) a time series in order to evoke Burg's algorithm. There is no need to define functions with slightly different names (such as "rburg" and "cburg") which essentially perform the same operation but for different types of time series.

Second, message passing allows us to construct an *abstraction barrier* between usage and implementation. For example, when we pass the message "burg" to a object belonging to the class **real-time-series** in the course of a data analysis, we really don't care about the implementation details. The system designer may well have implemented Burg's algorithm for real-valued series by making use of the complex version of the algorithm, but this shouldn't matter to the user. Conversely, if the system designer decides that the use of the complex algorithm for real-valued series is too inefficient, he or she can change this implementation detail without disrupting users. This scheme guarantees the user a certain response from a certain message, yet gives the designer the option of changing the underlying details.

3.3 Inheritance

Inheritance is a mechanism in OOP which allows us to construct new classes of objects based upon modifications to existing classes. Again we use an example from P_ITSS_A for illustration. The objects in the class **ordered-x-y-pairs** can be used to describe some of the properties of a time series sampled at arbitrary points in time. Two messages which can be sent to an object of the class **ordered-x-y-pairs** are "mean" and "mean-time." The first returns the average value of the time series (i.e., the average of the values in the vector bound to the slot *y-values*), and the second, the average time at which the observations were collected (i.e., the average of the values in the vector bound to the slot *ordered-x-values*).

Now the class **real-time-series** is intended to represent real-valued time series sampled over an equally spaced grid. The class is obviously quite similar in some respects to **ordered-x-y-pairs**. We may take advantage of this similarity by defining **real-time-series** such that it inherits all of the slots and ways of handling messages of the class **ordered-x-y-pairs**. As an example, sending the message "mean" to an object of the class **real-time-series** would make use of the slots and message handling inherited from **ordered-x-y-pairs**. We are free, however, to define additional slots and way of handling both new and inherited messages for our new class. These would express the difference between the intended use for the two classes. For example, we could define the slots *sampling time* and *first-time-value* to replace the functionality of the slot *ordered-x-values* in **ordered-x-y-pairs**. The values of these two new slots can be used to generate all the time values for a time series sampled over an equally spaced grid of times — there is no need to use the vector *ordered-x-values* explicitly. Likewise, we can redefine how the message "mean-time" is handled by **real-time-series** so that it computes it using its two new slots and the slot *number-of-pairs* inherited from **ordered-x-y-pairs**.

The advantage of using inheritance is that it allows us to construct rather complicated objects out of simplier one in a way that clearly expresses the differences between related classes. This is a quite useful way of a modifying and extending a large software system.

4. DESIGN OF P_ITSS_A

There are three major classes in P_ITSS_A — data objects, graph objects, and frame objects. The first represents various types of time series and the results of processing them; the second is used to construct the graphical output of P_ITSS_A on a bitmap display; and the third handles the user interface. Our discussion below of objects in these classes is far from exhaustive — we only describe a few of each kind to give the reader a feel for the organization of P_ITSS_A.

4.1 Data Objects

We have already described briefly three classes of data objects in Sections 3.1 and 3.3 — **ordered-x-y-pairs**, **real-time-series**, and **complex-time-series**. There are many others which are used to represent various other kinds of time series, such as **real-time-series-with-missing-values** (a real-valued time series sampled regularly but with missing observations) and **vector-time-series** (a vector valued times series sampled regularly). Each of these classes has slots (or inherits slots from component classes) for the actual values of the time series; the symbolic units for the time series values; the sam-

ling time and Nyquist frequency (for regularly sampled series); the network name of an ASCII file from which the values of the time series were read; various results from statistical computations (such as the sample autocovariance function); and so forth.

When a message is passed to a particular data object, the object responds by returning either a single value, a compound data structure, or another object. Examples of these for the class **real-time-series** are the message "mean" (which returns the average of the values in the time series); "acvf" (which returns a vector with the values of biased estimator of the autocovariance function and, at the same time, caches them in a slot in **real-time-series** for possible future use); and "burg" (which returns an object of the class **arima-model-object**). In the last case, the returned object can itself respond to messages — for example, an object in the class **arima-model-object** can respond to the message "sdf" in order to return to the user calculated values (over a specified grid of frequencies) of the spectral density function for the ARIMA model described by the object.

The various classes of data objects are intended to provide basic support for the purely computational aspect of interactive time series and spectral analysis. With just the classes and messages defined here, a user can carry out a data analysis by typing in Lisp expressions for evaluation by the Lisp reader; the expression will result in various messages being passed to various objects; and the user can assign the resulting values (or returned objects) to various symbolic names for latter use. This mimics completely the interaction that typically occurs in a interactive data analysis system, with the additional advantages of generic functions, support for complicated data structures, and a full-fledged programming language supported by the Lisp programming environment.

4.2 Graph Objects

Graph objects support two things: displays of results of various computations on a bitmap terminal; and interaction of the user with these graphs by means of a "mouse" and a keyboard. Slots in the class **basic-graph-object** provide for lists (or sets) of various other objects, each of which can response to the message "draw-yourself." These other objects describe various portions of a graph — the axes, the titles, mouse-sensitive regions (i.e., areas on the graph over which a click of the button on the mouse causes something to happen), plots of data, and so forth.

The basic way in which a bitmap graph is constructed in P_ITSS_A is by attaching drawable objects (i.e., those which know how to respond to a "draw-yourself" message) to an object of the class **basic-graph-object**. Once this has been done, the user sends a "draw-yourself" message to that object, and it in turn sends "draw-yourself" messages to the objects in its list of drawable objects. This makes it relatively easy to extend P_ITSS_A to create specialized graphs — the user need only define an appropriate class of objects with slots to support the desired features and an appropriate definition for the "draw-yourself" message.

An important point to note is that drawable objects are truly separate entities in P_ITSS_A. Thus a drawable axis object can actually be on the list of drawable objects for several different objects of the type **basic-graph-object**. This allows us to maintain consistency in the visual representation of related graphs. For example, we might have several plots of different spectral estimators for the same time series. If the objects which represent these plots each has the same drawable axis object for the vertical axis, then changes to this object (say, in its maximum axis value) can be made to propagate automatically to all graphs of which this axis object is a component.

4.3 Frame Objects

Frame objects are designed to support the user interface in P_ITSS_A. An experienced user could carry out an interactive data analysis by just creating various data objects and graph objects and sending messages to them. Since these are all implemented in Lisp, he or she could define "on the fly" new functions (or messages) to investigate a data set thoroughly as new ideas for exploring data arise.

However, there is also a need in a statistical analysis system to carry out fairly routine procedures (particularly for the novice user). Frame objects allow us to support these by defining a useful user interface for particular procedures; by packaging together sequences of calls to Lisp functions appropriate for a particular type of analysis; by storing important values returned from these calls in slots in the frame object for later use; and by causing the display of bitmap graphics to occur by attaching appropriate drawable objects to objects of the class **basic-graph-object** and sending these latter objects the message "draw-yourself." All of these actions occur when a frame object receives the message "do-frame."

Each class of frame objects thus supports only

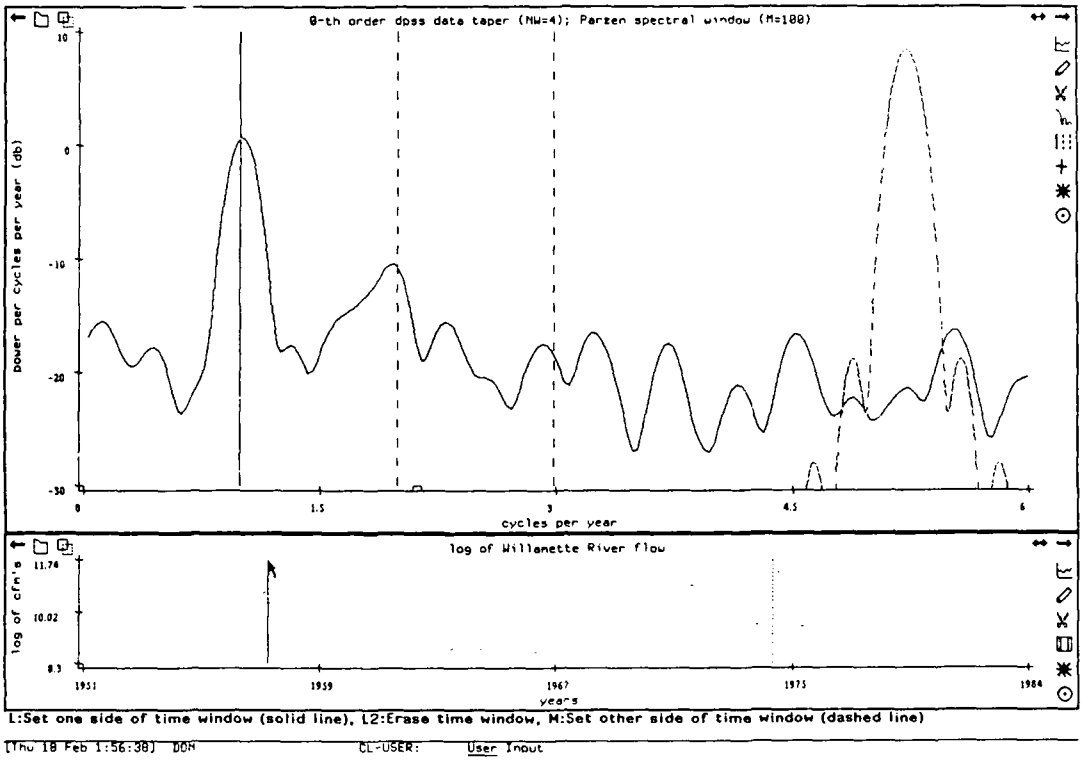


Figure 1. *Screen dump of the bitmap of a monitor of a Symbolics Lisp Machine showing the results of sending a "do-frame" message to an object of the class* **windowed-periodogram-frame**. *The screen dumps shows two distinct plots — the upper one shows a line plot of an estimate of the spectrum versus frequency for some flow data from the Willamette River in Salem, Oregon, while the lower plot shows a point plot of the corresponding time series.*

one type of statistical procedure. Some examples of these classes are **windowed-periodogram-frame** (described in the next section) and **autoregressive-sdf-frame**. The code which handles the "do-frame" message for each frame is in effect a small script of actions for carrying out common procedures in time series and spectral analysis. As such, they are good places to look for ideas on implementing new procedures. If a user has an idea for a particular type of procedure which is close to an existing frame procedure in P_ITSS_A, he or she can often modify that existing frame (possibly by using the inheritance mechanism of OOP) to create a new frame class which is tailored to the new procedure.

5. AN EXAMPLE

In this section we give an example to clarify some of the ideas in Section 4. The example centers around Figure 1, which is a screen dump of the monitor of a Symbolics Lisp Machine. This display shows the result of sending a "do-frame" message to an object of the class **windowed-periodogram-frame** (hereafter referred to as the frame object). This message comes with a single argument, which must be an object of either the class **real-time-series** or the class **complex-time-series** (hereafter called the time series object). In the present example, the time series object represents the log of the average monthly flow of the Willamette River at Salem, Oregon from 1951 to 1984 (the dots on the bottom plot of Figure 1 show a plot of this series versus time).

After the frame receives the "do-frame" message, it presents a menu of options to the user. These concern, among other things, prewhitening, data tapering, type of spectral window, and associated window parameter (see Priestley (1981) for a discussion of the technical details of spectral analysis). After the user specifies these options, the frame object sends the necessary messages to the time series object to calculate a windowed periodogram spectral

estimate. The time series object returns one of the class **windowed-periodogram-spectral-object** (hereafter the spectral object) to the frame object — the spectral object represents the spectral estimate and associated information (bandwidth, variance, degrees of freedom, etc.) and is bound to a slot in the frame object for future reference.

The frame object next makes use of two objects of the class **basic-graph-object**, which we hereafter call the time graph object and the spectral graph object. The time graph object is used to create a bitmap plot of the time series associated with the time series object, while the spectral graph object does the same for a plot of the windowed periodogram spectral estimate. To set this up, the frame object attaches the time series object (spectral object) to the list of drawable objects in the time graph object (spectral graph object). The frame object also attaches axis objects, title objects, and mouse-sensitive region objects to the appropriate lists in each graph object.

After objects are attached, the frame object sends a "draw-yourself" message to both graph objects. Each graph object in turn sends a "draw-yourself" message to each of the objects in its list of drawable objects. The results are a large plot of the spectral estimate versus frequency (the solid line in the upper part of Figure 1) and a smaller plot of the time series versus time (the dots in the lower part).

When a graph object sends a "draw-yourself" message to a mouse-sensitive region object, a small icon is drawn in the margin of the plot. For example, the right margins of both plots in Figure 1 each have a vertical stack of icons. The top four icons in each stack are the same (a right arrow, a graph icon, a pencil icon, and a scissors icon), as are the bottom two (an asterisk icon and a button icon). The upper spectral plot has three additional icons (a kernel icon, a harmonics icon, and a variance/bandwidth crossbar icon), while the lower time plot has only one (a shaded window icon). Once the plots have been displayed on the bitmap screen, the user can move the mouse cursor (an arrow pointing in the 11 o'clock direction) until it is over a particular icon and click the mouse button to cause a particular mouse state object to be activated on the corresponding graph object. This mouse state object is then used to interpret any mouse clicks the user makes over any portion of the graph which is not part of a mouse-sensitive region.

Three examples of the interactions possible via these mouse-sensitive regions are shown in Figure 1. If the user clicks on the kernel icon (the one below the scissors icon on the spectral plot), a mouse-state object is activated which allows the user to draw the kernel $K(\cdot)$ associated with the windowed periodogram spectral estimate $\hat{h}(\cdot)$, where the kernel appears by the relationship

$$E\{\hat{h}(\omega)\} = \int_{-N_f}^{N_f} K(\omega - \lambda)h(\lambda)\,d\lambda\,;$$

here N_f is the Nyquist frequency and $h(\cdot)$ is the true spectral density function ($K(\cdot)$ depends upon the data taper and the spectral window). The user specifies where $K(\cdot)$ is to be drawn by pointing and clicking the mouse button — this defines where the top of the central lobe of $K(\cdot)$ is to be plotted. (Internally, the drawing is accomplished by attaching a kernel object to the list of drawable objects in the spectral graph object and sending it a "draw-yourself" message.) In Figure 1 we placed $K(\cdot)$ at the upper right-hand part of the spectral plot (shown as a dashed line). Subsequent clicks allow the user to relocate $K(\cdot)$ anywhere else in the central plotting area of the spectral plot. This allows the user to visually assess two important aspects of the spectral estimate. First, if there are sharp features in $h(\cdot)$, these will essentially cause the central lobe of $K(\cdot)$ to be traced out. This does occur in Figure 1 — there is a sharp feature located at 1 cycle per year (corresponding to the annual flow cycle in the Willamette River), and the spectral estimate in that region has the same shape as $K(\cdot)$ (as could be seen quickly by relocating $K(\cdot)$ there). Second, the height of the sidelobes relative to the main peak as compared to the observed dynamic range in $\hat{h}(\cdot)$ is a good visual indication of whether there may be significant bias in $\hat{h}(\cdot)$ due to window leakage. This is evidently not a problem in our example — the sidelobes of $K(\cdot)$ decay rapidly compared to the dynamic range observed in $\hat{h}(\cdot)$.

For our second example, we describe the use of the harmonics icon (the one below the kernel icon). This icon is used to mark (using the mouse) the location of a fundamental frequency and a certain number of its harmonics. After the user clicks the mouse over this icon, a mouse state object is activated which first displays a small menu to query the user about the number of harmonics to be drawn. In the example in Figure 1, we requested 2 harmonics. From then on, the mouse state object interprets a mouse click as being the location at which the user wishes to have a marker drawn indicating a fundamental frequency. This is drawn in Figure 1 as a solid vertical line at 1 cycle per year on the spectral plot; the corresponding first two harmonics of

this frequency are indicated by vertical dashed lines (again, this is accomplished internally by attaching a drawable harmonics object to the spectral graph object and sending it the message "draw-yourself). This allows the user to determine whether prominent features on a spectral estimate are harmonically related and can thus be attributed to a periodic phenomenon in the time series. (If one or more the harmonics is higher than the Nyquist frequency, its alias in the interval $(-N_f, N_f)$ is drawn as a vertical dashed line with a smaller dash size.)

Our third example shows how extracting a subseries from a time series can be done on PITSSA (this can be useful to investigate whether any visual differences in the time plots of various subseries map over into the frequency domain). This makes use of the shaded window icon immediately below the scissors icon on the lower plot. Once the user clicks over this icon with the mouse, a mouse state object is activated which allows the user to place two vertical markers on the time series plot by pointing and clicking with the mouse. These markers are shown in Figure 1 as a vertical solid (dashed) line near 1957 (1974). Once these two markers are in place, the user can request that the windowed periodogram frame calculations be repeated using only the subseries defined by the markers instead of the original series. This can be done by positioning the mouse cursor over the button icon either on the time series plot or the spectral plot (the bottom icon in the right-hand column on either plot) and clicking the mouse button — this causes a "do-frame" message to be send to the frame object with qualifiers which cause it to use the subseries instead of the full series as data.

6. CONCLUSIONS

We conclude by reconsidering the three questions posed in Section 1. First, PITSSA was specifically designed to make systematic use of the graphical interface possible with a modern computer workstation. It accomplishes this through the use of mouse state objects, drawable mouse-sensitive region objects, and other drawable objects. These objects share a common interface in PITSSA through a set of common messages to which they can respond. This uniformity makes it clear how to define new classes of objects to extend the system gracefully.

Second, the use of frame objects allows us to define a high-level graphical interface for PITSSA in terms of more fundamental operations on data objects and graph objects. The response that we have gotten from students who have used the system convinces us that this interface is useful for novices. Our hope is that the overall design of PITSSA is transparent enough that more sophisticated users (other than ourselves) can augment and modify it at will (this level of usage remains to be tested). There are really no constraints imposed on a sophisticated user of the system other than those imposed by Lisp itself — in fact, PITSSA may be regarded as simply a nefarious plot to get innocent users interested in the programming potential of the Lisp environment itself!

Third, we hope that the three simple examples in Section 5 convince the user of the usefulness and power of interactive graphics in time series and spectral analysis (and other areas of data analysis). There are several other interesting examples from PITSSA which we plan to discuss in future articles. There is also much more work to be done in this fruitful area before we exhaust the potentials for improving interactive data analysis pointed to by McDonald and Pedersen (1985a, 1985b, 1988).

7. ACKNOWLEDGEMENTS

This research was sponsored by the Office of Naval Research under contract numbers N00014-87-K-0441 and N00014-81-K-0095. The Macintosh II implementation of PITSSA which we mention briefly is an on-going project sponsored under contracts from the Naval Observatory and the Naval Research Laboratory. (The name PITSSA stands for "Program for Interactive Time Series and Signal Analysis" and is pronounced like the popular Italian dish. Earlier versions of this program were called TSA (Time Series Analysis), an acronym we abandoned after seeing three commercial products with that same name at a recent statistics conference! Although we know of no other PITSSA's in existence, we have followed the example of the author of TEX (Knuth 1984) and lowered our vowels in an attempt to create a unique name this time around.)

REFERENCES

Becker, R. A., and Chambers, J. M. (1984), *S: An Interactive Environment for Data Analysis and Graphics*, Monterey, California: Wadsworth.

Becker, R. A., and Chambers, J. M. (1985), *Extending the S System*, Monterey, California: Wadsworth.

Burg, J. P. (1975), "Maximum Entropy Spectral Analysis," unpublished Ph.D. thesis, Stanford University, Department of Geophysics.

Knuth, D. E. (1984), *The TEXbook*, Reading, Massachusetts: Addison Wesley.

Marple, S. L. (1987), *Digital Spectral Analysis with Applications*, Englewood Cliffs, New Jersey: Prentice-Hall.

McDonald, J. A. (1986), "Antelope: Data Analysis with Object-Oriented Programming and Constraints," in *Proceedings of the Statistical Computing Section, American Statistical Association*, 1–10.

McDonald, J. A., and Pedersen, J. (1985a), "Computing Environments for Data Analysis, Part 1: Introduction," *SIAM Journal on Scientific and Statistical Computing*, 6, 1004–1012.

— (1985a), "Computing Environments for Data Analysis, Part 2: Hardware," *SIAM Journal on Scientific and Statistical Computing*, 6, 1013–1021.

— (1988), "Computing Environments for Data Analysis, Part 3: Programming Environments," *SIAM Journal on Scientific and Statistical Computing*, in press.

Olford, R. W., and Peters, S. C. (1986), "Data Analysis Networks in DINDE," in *Proceedings of the Statistical Computing Section, American Statistical Association*, 19–24.

Priestley, M. B. (1981), *Spectral Analysis and Time Series*, London: Academic Press.

Steele, G. L. (1984), *Common Lisp: The Language*, Burlington, Massachusetts: Digital Press.

Stuetzle, W. (1987), "Plot Windows," *Journal of the American Statistical Association*, 82, 466–475.

Thomson, D. J. (1982), "Spectral Estimation and Harmonic Analysis," *Proceedings of the IEEE*, 70, 1055–1096.

INSIDE A STATISTICAL EXPERT SYSTEM: Implementation of the ESTES system

Paula Hietala, University of Tampere, Finland

ABSTRACT

In this paper we describe the implementation of a statistical expert system called ESTES. The system is intended to provide guidance for an inexperienced time series analyst in the preliminary analysis of time series. The ESTES system has been implemented on Apple Macintosh™ micro-computers using a combination of Prolog and Pascal languages.

Keywords: Statistical expert systems; Rules; Explanation capabilities

1. INTRODUCTION

Statistical expert systems are an interesting and novel area of statistical computing today (see e.g. Chambers (1981), Gale (1986a) and Hietala (1987)). However, the implementations of these systems are often outlined very cursorily and the reader is left unaware or in doubt of the methods employed as well as of the inner structure of the systems. The purpose of this paper, on the contrary, is to consider in more detail one implementation (of a statistical expert system called ESTES) in order to give a better insight into these popular systems.

The ESTES (Expert System for TimE Series analysis) system is intended to provide guidance for an inexperienced time series analyst in the preliminary analysis of time series, i.e. in detecting and handling of seasonality, trend, outliers, level shifts and other essential properties of time series. In the preliminary analysis of time series it is usually the case that an expert time series analyst detects the essential features of a time series just by examining its graphical representation and autocorrelation function, without any complicated calculations. Even in the case of an inexperienced user he/she may have plenty of useful knowledge concerning the environment of the problem in question. With this in mind, the statistical knowledge in the system is organized so that the system tries to exploit as much as possible of the knowledge or experience that the user has about the specific time series being considered. However, if there exists a conflict between the results computed by the system and the knowledge elicited from the user, then the ESTES system sets out to carry out more extensive analysis and apply more sophisticated statistical methods. With this kind of organization we strive for minimizing the number of unnecessary reasoning and calculation steps.

The ESTES system has been implemented on Apple Macintosh™ personal microcomputers using Prolog and Pascal languages. In this paper we consider the overall implementation of the system. The design philosophy and user interface principles of the system are described in Hietala (1986). The organization of the knowledge base and the statistical methods employed in the system are given a detailed treatment in Hietala (1988).

2. STRUCTURE OF THE ESTES SYSTEM

The structure of the ESTES system and the communication between its principal modules is illustrated in Figure 1. The system consists of:

- *a main module* which takes care of communication between other modules,
- *a statistical knowledge base* which comprises the knowledge about time series analysis,
- *an inference engine* which employs the knowledge in the statistical knowledge base,
- *a user interface module* which interacts with the user,
- *a graphics module* which displays graphical results,
- *a time series generation module* which generates example time series, and
- *numerical computation modules* which calculate all numerical results for the other modules of the system.

The numerical computation modules have been implemented in Pascal, all the other modules in Prolog.

Next we briefly describe each of the modules.

2.1. User interface module and graphics module

The user interface of the ESTES system is especially designed for an inexperienced user (see Hietala (1986)). The system is highly interactive: the user interacts with the system using pull-down menus, dialog windows, overlapping and transferable data windows, with a mouse as a pointing and selection device. Figure 2 illustrates the Macintosh-like user interface of the system. Whenever possible both *numerical* and *graphical displays* of the data and statistics (for example, autocorrelations and partial autocorrelations calculated from the data) are offered to the user. Also the shape parameter of graphical displays (the ratio of height and width of a figure) may be chosen by the user. For example, in Figure 2 we have a graphical display of data (see the window "Time Series: x"). If the user wants to change the shape

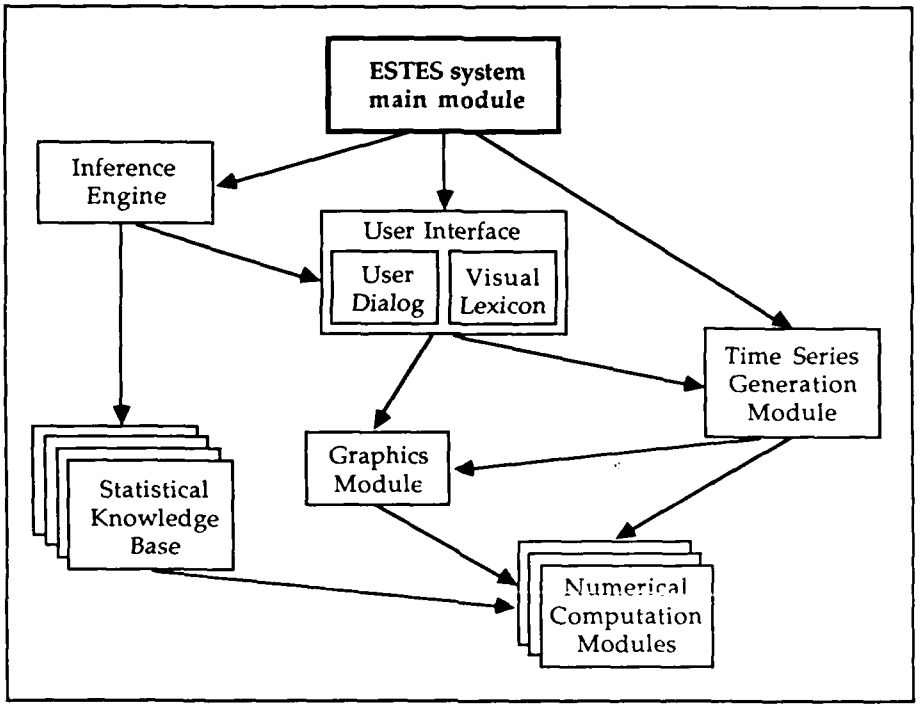


Figure 1. The structure of the ESTES system.

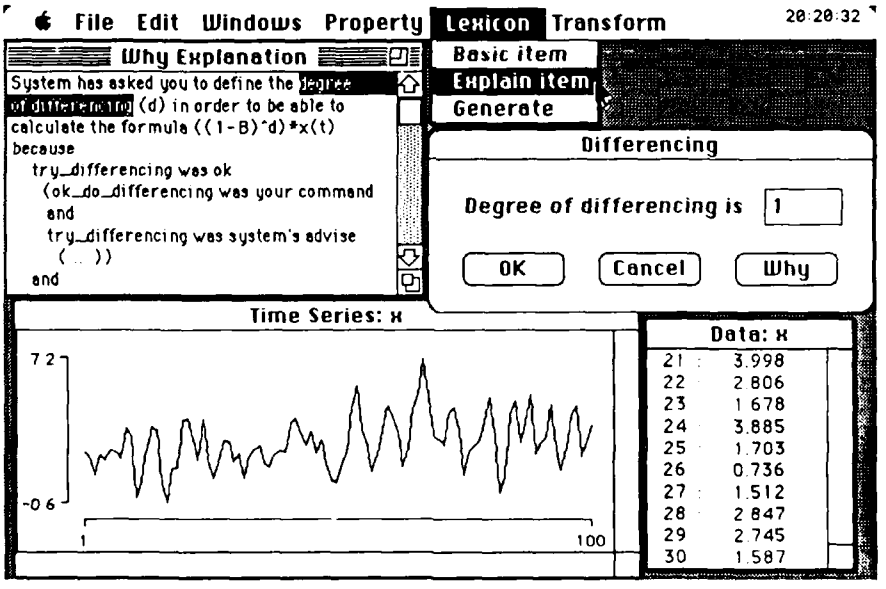


Figure 2. An example screen of the ESTES user interface

parameter of this display, he/she activates the graphical window and chooses a new shape parameter by using "Window details" command from the "Windows" menu. Also, the user may activate with a mouse any individual point in the graphical display: the value of time series and and time point are shown in the display. Moreover, the numerical display of the data (see the window "Data: x") is scrollable and editable. Changes and insertions in the data window are immediately seen also in the graphics window.

The visual lexicon of the system (see the Lexicon menu) gives support to the user during his/her work. This is similar to the lexicon concept of Gale (1986b); however, in our system the lexicon illustrates the definition of unknown statistical terms *graphically* and explains the meaning of the terms used by the system. For example, if the user asks for an explanation for the term "trend", the system produces both a graphical representation of a time series with a trend and also a textual explanation of the term "trend". The visual lexicon contains a rather small amount of precomputed information for the graphical representation; the reason is that we strive for *a dynamic lexicon*. By this we mean the ability of the system to produce several different examples of a phenomenon in question. We have implemented this feature by including generation rules to the lexicon instead of example data sets.

Let us consider our example situation in Figure 2 a little more closer. Let us assume that the user wants to remove trend from time series x by applying non-seasonal differencing. Therefore the system has inquired about "the degree of differencing" (see *the Differencing window:* our system's suggestion for the degree of differencing is 1). Next, the user likes to know why the system asks this fact (he/she chooses the soft button "why" in the Differencing window). After that, the user can read the explanation from *the Why explanation window*. However, the user wants still more information about the term "degree of differencing", so he/she selects the term in question by marking from the Why explanation window and then requests the system to explain this term through its visual lexicon (the user chooses the corresponding action from the Lexicon menu).

The ESTES user interface module (e.g. graphics windows and menus) has been implemented using LPA MacPROLOG™ compiler (see Clark et al. (1988)) and its advanced graphics tools. Interestingly, Prolog seems to be well-suited for this task; only a few numerical calculations in the user interface module have been implemented in Pascal for the sake of convenience.

2.2. Knowledge base and inference engine

A knowledge base for storing the expert's knowledge of a problem domain and *an inference engine* for inferring solutions and explaining system's actions are at the very heart of any expert system: this is also the case with statistical expert systems. Next we briefly describe the principles employed in the ESTES system in implementing these two components. A more detailed account of these matters can be found in Hietala (1988).

There are several ways of representing knowledge in the knowledge base. We have selected *if-then rules* for representing knowledge concerning properties of time series and their handling. Rules in our system are either of form: `RuleName: if condition_A then conclusion_B`, or of form: `RuleName: if condition_A then action_C`. This kind of rules are easily expressed in Prolog: they are legal Prolog clauses provided we define appropriate Prolog operators (e.g. ':', 'if', 'then'). The condition and action parts of a rule usually include also invisible calls to Pascal procedures (see Section 2.4 for a more detailed discussion of the interplay between Prolog and Pascal).

The knowledge base of the ESTES system has been organized so that the selection of a class of statistical methods will be determined using a hierarchy of criteria, i.e. according to

(1) the property being considered,
(2) the granularity of analysis process (whether we are performing initial or more extensive analysis),
(3) the goal of the analysis process (detecting or handling the property in question), and
(4) the knowledge possessed by the user about the specific property as well as on his/her general knowledge about time series (the background of the user may vary from a student to an expert).

Within the chosen class of statistical methods, the final selection will be made according to the power of the methods, i.e. the most powerful method available is selected first.

Although the Prolog language is itself *an inference engine* it is not sufficient for our purposes. We do not use Prolog's own trace facility but have built an interpreter on top of Prolog. This interpreter manages the reasoning process of the ESTES system: it interacts with the user during the reasoning process and also after it. For example, after the system has asked the user about some information concerning the time series the user can ask a *'why'* question ("Why does the system inquire this fact?"). Also, after the system has completed its reasoning process the user may ask *'how'* questions ("How has the system reached this conclusion?"), see e.g. Bratko (1986). Our system's reply to why and how questions consists of displaying a user-friendly form of its inner inference chain with explanations and justifications of those methods that are used inside the chain. In addition to the textual explanation, the system's answer can contain displays of graphical results.

2.3. Time series generation module

Graphical examples in time series analysis (as well as in other branches of statistical analysis) can be very illuminating and instructive for an inexperienced analyst. Besides for this purpose *the time series generation module* of our system is also utilized by the lexicon mechanism when producing examples of graphical representation of statistical issues inquired by the user. A third use of the time series generation module is in the development phase of a statistical expert system: the developer can generate various test series (that otherwise would be difficult to obtain) and examine the system's behaviour with respect to these test series.

So, the user first generates his/her own example data (for example, a specific time series with a property he/she is interested in) using the generation feature and then he/she can examine this data (time series) with the help of the system. Thus the generation feature alleviates the learning of preliminary time series analysis: the user can very easily get acquainted with typical time series and their properties.

The ESTES system applies ARIMA models in the generation process but the user does not necessarily need to have knowledge about ARIMA models. He/she only describes the properties which he/she wants to be embedded in time series and the system chooses the appropriate model. But if the user wants, he/she can also define a precise model structure for the time series generation process.

The actual computation in the time series generation module is implemented in Pascal, employing the numerical computation modules of our system.

2.4. Numerical computation modules

The ESTES system like other systems performing statistical calculations demands quite heavy *numerical computation power*. Prolog is not designed for numerical but for symbolic computation, so for efficiency reasons we have not used Prolog for numerical computation. Computational components of statistical expert systems usually employ some existing statistical software package or are programmed in an ordinary procedural language, such as Pascal or C. Unfortunately we did not find any sufficiently flexible existing statistical software package for the implementation, so the use of a procedural language (in our case, Pascal) for all numerical computation was necessary.

Figure 3 illustrates the interplay of Prolog and Pascal languages. On the left we have a fragment of

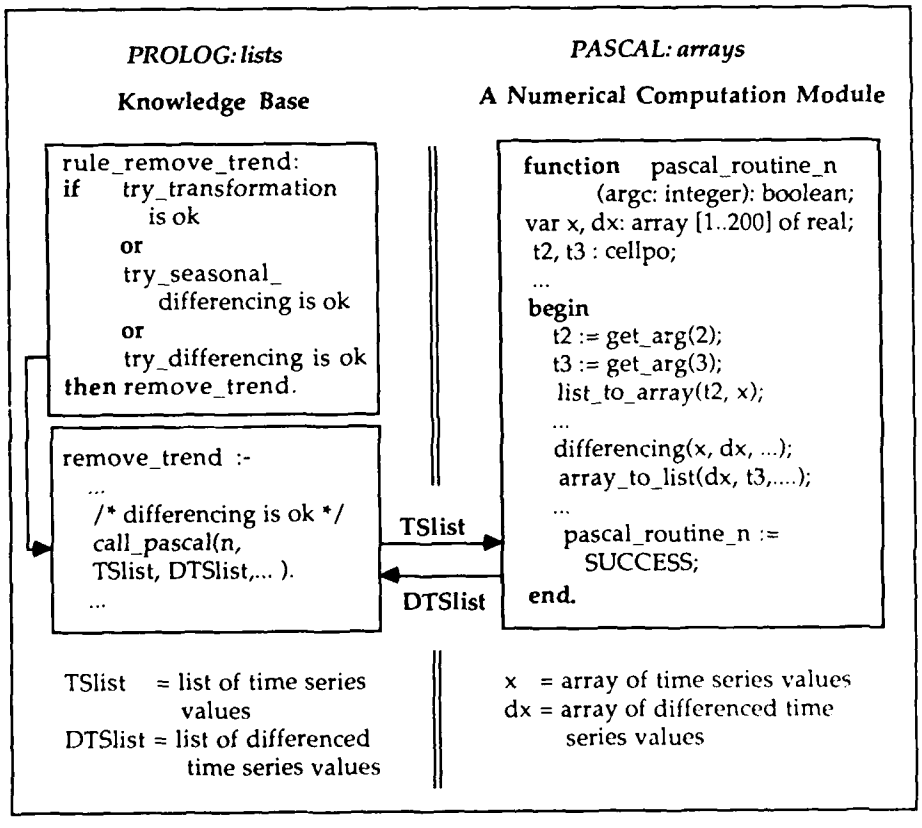


Figure 3. The interplay between Prolog and Pascal languages.

the knowledge base, coded in Prolog and on the right one of the numerical computation modules, coded in Pascal. Let us assume that in the knowledge base the rule 'rule_remove_trend' is selected because the condition 'try_differencing is ok' is true. So the action 'remove_trend' is executed next. The body of this action contains a special Prolog predicate 'call_pascal' which has as one of its parameters the list of time series values. This list is passed to the corresponding Pascal function, which converts the list to an array and then carries out the actual differencing. After that the results are returned as a list to Prolog. To the user of the system, however, the interplay of these two languages is hidden.

3. CONCLUDING REMARKS

The ESTES system is an experimental research vehicle for studying the use of artificial intelligence (AI) techniques in producing statistical expert systems. Our system has not yet been tested in real-life situations, because its current knowledge base is too small. In the near future our main emphasis in the development of the system will be in deepening its domain knowledge concerning preliminary time series analysis.

However, we think that our system rather nicely embodies the two faces of statistical expert systems, i.e. *the deductive component* (usually programmed using an expert system shell or an AI programming language, such as Lisp or Prolog) and *the computational knowledge component* (which usually employs some existing statistical software package or procedures programmed in an ordinary procedural language, such as Pascal or C). In our opinion, this "dynamic knowledge base" (the computational knowledge component outlined above) is very typical for statistical expert systems.

In our case, the use of a combination of the languages Prolog (in the deductive component) and Pascal (in the computational knowledge component) turned out to be a very suitable way of implementing a statistical expert system.

REFERENCES:

Bratko, I. (1986). *Prolog Programming for Artificial Intelligence*. Addison-Wesley, Wokingham, England.

Chambers, J.M. (1981). Some thoughts on expert software. *Computer Science and Statistics: Proceedings of the 13th Symposium on Interface*. Springer-Verlag, New York, NY, 36-40.

Clark, K.L., McCabe, F.G., Johns, N., and Spenser, C. (1988). *LPA MacPROLOG Reference Manual*. Logic Programming Associates Ltd, London, England.

Gale, W.A. (ed.), (1986a). *Artificial Intelligence & Statistics*. Addison-Wesley, Reading, MA.

Gale, W.A. (1986b). REX Review. In Gale (1986a), 173 - 227.

Hietala, P. (1986). How to assist an inexperienced user in the preliminary analysis of time series: First version of the ESTES expert system. *Proceedings in Computational Statistics (COMPSTAT) 1986, 7th Symposium held at Rome 1986*, Physica Verlag, Heidelberg, 295-300.

Hietala, P. (1987). Statistical expert systems for time series analysis. Paper presented at *The First Conference on Statistical Computing (ICOSCO-I)*, Cesme, Turkey, 30 March - 2 April, 1987.

Hietala, P. (1988). Inside a statistical expert system: Statistical methods employed in the ESTES expert system. Paper accepted to the *Computational Statistics (COMPSTAT) 1988, 8th Symposium*, Copenhagen, Denmark, 29 August - 2 September, 1988.

THE EFFECT OF MEASUREMENT ERROR IN A MACHINE LEARNING SYSTEM

David L. Rumpf Mieczyslaw M. Kokar
Department of Industrial Engineering and Information Systems
Northeastern University, Boston, MA

ABSTRACT

This paper deals with the problem of reasoning about conceptualizations (sets of relevant parameters) of physical processes. The problem is discussed in the context of the COPER discovery system. COPER conjectures parameters characterizing physical processes and the functional relationships among them. The COPER system utilizes the idea of changing representation base to determine the arguments of invariant functional descriptions. It must handle two kinds of uncertainty - about relevance of parameters, and measurement error. A statistics/probability approach has been used to estimate the effect of measurement error in the COPER system. The partially adequate results of this approach are presented. Alternative approaches to the measurement error problem will be suggested.

INTRODUCTION

The process of discovery of a physical law involves reasoning based upon experimental data obtained from observations and measurements. The measurements are never perfect, they always include both the essential information about the behavior of the physical system and some noise. A discovery system must be able to perform reasoning on such noisy data and extract the causal relationships. The indication for existence of such a relationship is some regularity in the data. This regularity must be described on at least two levels: some conceptualization (a set of concepts) must be defined, and then a relationship should be described in terms of these concepts. This is a very difficult task. One of the reasons for this is that a discovery system must deal with two kinds of uncertainty:

- related to the lack of knowledge on whether the parameters the system is measuring are all the relevant parameters,
- caused by noise in the input data.

This paper reports how these two problems of uncertainty are handled by the discovery system called COPER [Kokar 1986a, 1986b].

Two approaches to discovery of regularities in measurement data can be distinguished - let us call them piece-incremental and global. In the piece-incremental approach a conjecture is made after getting any single new piece of data. In the global approach a conjecture is made after all data has been collected. Between the boundaries delineated by these two approaches there is room for all kinds of mixed strategies - repetitions of collecting of some amount of data and drawing an inference. The mixed strategies can be viewed as combinations of the two. After we collect some amount of data we make some global reasoning, then the conclusion can be treated as one piece of information which can be input to the piece-incremental reasoning system. In this paper we concentrate on the global reasoning approach.

We introduce some measure, which is the fundamental tool for making inferences in the global approach. Measures are functions which assign numeric values to sets of observations. The measure we introduce in this paper assigns a numeric value to a conceptualization of a physical process. It is based on the predictive power of a conceptualization - the better the prediction the lower the value of the measure. The utilization of such a measure in the process of deriving a physical law from observational data is obvious - the system generates conceptualizations of a physical process, applies the measure to them, and selects the conceptualization for which the value of this measure takes its minimum.

Such a statement of the problem might suggest that the system generates a model of the process and then tests its predictive power. This could be called a "traditional approach". The drawback of such an approach is that we are not able to assess which part of the model is responsible for the wrong predictions. In the approach presented in this paper we construct a measure which:

- is able to assign blame/credit to particular parameters in the model,
- does not require postulating a functional dependency describing the model (due to the fact that the definition of this measure utilizes the principle of similarity we call it the "similarity measure").

THE SIMILARITY MEASURE

In the process of constructing the similarity measure we make use of some syntactic properties of physical laws. We consider here the laws which are represented by some functional formulas (or algorithms). The arguments of these functions are so called "dimensional quantities", i.e., a numeral followed by several "units" with some exponents. For instance, a physical quantity of "velocity" is expressed as:

$$V = x_v kg^0 m^1 s^{-1}.$$

The functions describing physical laws must fulfill some constraints. For instance the function Y = 3 m + 5 kg does not have an interpretation in the language of physics, thus it should be disallowed. The constraints guarantee that by performing some syntactic operations on a representation of a physical process we do not generate some objects which are not interpretable in the domain. The constraints are captured by the requirement of dimensional invariance of functions representing physical laws with respect to the change of the representation. Formally the invariance is represented as:

$$F(TX_1,\ldots,TX_n) = TF(X_1,\ldots,X_n),$$

where F is a functional formula, $X_1,...,X_n$ are physical parameters (dimensional quantities), and T is a transformation of the representation. The discussion of the syntactic properties of physical laws is beyond the scope of this paper, an interested reader is asked to refer to the subject's literature (e.g., [Whitney, 1968], [Birkhoff, 1960], [Drobot, 1953], [Kokar, 1981, 1985]). This problem falls into the field of dimensional anlysis.

In the theory of dimensional analysis a theorem exists (called sometimes the Pi- theorem), which says that any function

$$Z = F(A_1,...,A_m,B_1,...,B_r),$$

which is dimensionally invariant, can be represented as a (new) function

$$Z = f(Q_1,...,Q_r)A_1^{a_1} \cdots A_m^{a_m},$$

where $Q_1,...,Q_r$ are the new parameters constructed out of the initial parameters according to the following formula:

$$Q_j = B_j / A_1^{a_{j1}} \cdots A_m^{a_{jm}}.$$

Dimensional analysis gives the rules for both the partitioning of the set of arguments into the A- and B-arguments, and for calculating the values of all the exponents in the above formulas.

Suppose a physical process is fully characterized by the physical parameters $A_1,...,A_m,B_1,...,B_r$, i.e., that the value of some characteristic of the process Z can be uniquely determined by these parameters (functionality).

By an "instance" we will mean one measurement of the physical process. Formally an instance R can be represented as an m+r+1-tuple:

$$R = <A_1,...,A_m,B_1,...,B_r, Z>$$

of the values of the parameters. When carrying out experiments with a physical process we obtain a collection of instances usually represented as a table. For any instance we can calculate the values of the Q-parameters, $Q_i(R)$.

Two instances R' and R'' are called "similar" when the following relationship holds:

$$Q_1(R') = Q_1(R''),...,Q_r(R') = Q_r(R'').$$

The similarity relation is an equivalence relation on a set of measurements M. Given a set of instances (measurements) M, a conceptualization $C = \{A_1,...,A_m,B_1,...,B_r,Z\}$, the similarity relation partitions the set M into equivalence classes; we call these classes "similarity classes".

The formula for Z can be transformed into

$$f(Q_1,...,Q_r) = Z / A_1^{a_1} \cdots A_m^{a_m} = Q_z.$$

Given a conceptualization, C, and a set of instances, M, the similarity measure SM(C,M) can be calculated in the following steps:

(1) Using dimensional analysis determine the forms of the monomials $Q_1,...,Q_r$.

(2) Partition the set of instances M into similarity classes.

(3) Using the above formula for f calculate the value of this function for each instance in M. Note, that to determine these values we do not need to make any assumptions about the form of the function f, we calculate them using the right side of the above formula.

(4) For each similarity class calculate the mean value of f.

(5) SM(C,M) is the mean value of all absolute differences between the calculated values of the function f and the mean values of f for every similarity class.

PROPERTIES OF THE SIMILARITY MEASURE AND ITS USE IN REASONING

The similarity measure defined in the previous section has some very useful properties. The most important property of this measure can be summarized in the following statement.

> If a physical process is fully characterized by the conceptualization $C = \{A_1,...,A_m,B_1,...,B_r,Z\}$, i.e., Z functionally depends on the remaining parameters, then for any set of instances M of this process the value of the function f (or Q_z) is constant for any similarity class, and consequently, the value of the similarity measure SM(C,M) is equal to zero.

To prove this property of the similarity measure let us first notice that $f(Q_1,...,Q_r)$ must be constant on a similarity class. This is a consequence of both functionality of the relation f, and of the definition of a similarity class (a class is determined as a set of instances for which all the Q's are constant). In such a case the mean value of this function is equal to the (constant) value of the function, and thus the difference must be equal to zero. This is true for each single similarity class. The similarity measure is defined as the mean value of the absolute difference from mean value of f for all the classes, therefore the similarity measure must be equal to zero.

The contraposition of this property says that if the value of the similarity measure is not equal to zero then the dependency of Z on the conceptualization C is not functional. This means that some of the parameters in the conceptualization are missing. If a parameter B_k is missing from our considerations, then as a consequence, a respective Q_k parameter is missing too. It means that in our definition of similarity classes one of the constraints, $Q_k = B_k / A_1^{a_{k1}} \cdots A_m^{a_{km}} = \text{constant}$, is not taken into account.

The use of this measure is straightforward. The system can search for a conceptualization by adding one parameter at a time to it and calculating the value of the similarity measure. If the value of the similarity measure improves (significantly) then the theoretical parameter is included into the conceptualization, otherwise it is ignored and the search continues. The search stops when the value of the similarity measure is close (enough) to zero. One of the very important features of this algorithm is that a physical parameter does not need to be varied in the set of measurements M in order to be judged.

The similarity measure gives us some means to handle uncertainty about the set of parameters characterizing a physical process under investigation. Even assuming no noise in the measurement data we face the problem of how much is enough. As was pointed out in the above paragraph, at least two questions need to be answered: what does the similarity measure improves "significantly" mean, and what is close "enough" to zero?

A further complication is introduced by the presence of noise in the measurement data. Deciding if the significance measure has improved "significantly" or if it is close "enough" to zero can be confounded by the presence of measurement error. A lower value of the similarity measure could mean that the theoretical parameter just added should be included. Alternatively, it could indicate improved precision in the measurement data.

To answer these questions requires use of one or more of the models of approximate reasoning, e.g., probability/statistics, rough sets, fuzzy sets, Dempster-Shafer, and so forth. Our initial analysis, presented below, attempts to use the probability/statistics approach to evaluate the effect of noise in the measurement data.

STATISTICAL INTRACTABILITY OF THE COPER DECISION PROBLEM

For the COPER system described above, one knows that a complete set of parameters has been found when the similarity measure is zero. In a world of exact measurements, such a result would be unambiguous. However, measurement errors exist for physically measured quantities. Thus, the parameters (variables) which COPER considers as candidates for the physical law under investigation have uncertain value. It follows that the completeness decision is not necessarily straightforward. How constant does Q_z have to be? When is the observed non-zero value of the similarity measure due to measurement error of the parameters and when is it due to missing parameters?

Our application of probability/statistics to this problem attempts to answer the following question. Given information on the distribution of errors for the physical entities, what is the resulting error distribution for Q_z. We will assume that the measurement errors in the parameters ($A_1,...A_m,B_1,...B_r$) are normally distributed with a mean of zero and some known standard deviation. This is a common assumption for measurement error. Since Q_z is a function of the measured parameters, we are concerned with transmittal of variation through this functional relationship. Mathematical statistics would describe the situation as a function of random variables problem. There are analytic solutions to many such problems. As many readers might know, if one sums independent normal random variables, the result is also normal [Mendenhall, 1986]. If one multiplies independent random variables under conditions where no one random variable is dominant, the natural log of the result is a log normal distribution [Lewis, 1987]. However, the COPER problem is not limited to simple sums or products of parameters. Many of the transformations of interest require division. To describe the error distribution for Q_z, we must determine the variance of Q_z. However, it can be shown (see Appendix A) that the variance of (1/X), for X a normally distributed random variable, can not be found analytically. Thus division, one of the more common transformations used by COPER, immediately removes the problem from the realm of analytic solutions. We present below an approximate solution to the problem.

APPROXIMATE STATISTICAL SOLUTION TECHNIQUE

We have designed an simulation approach to determine the error distribution of Q_z under certain assumptions. For any particular analysis problem, there are a limited number of possible groupings of the A- and B-parameters into the Q- parameters. Each grouping imposes a particular functional form for Q_z. We define a representative problem which has several functional forms for Q_z. We then randomly assign errors to the parameters, and analyze the statistical variation in Q_z. If this statistical variation in Q_z can be shown to fit a known distribution, we argue that functional transformations of similar form will result in the same, known distribution of error variation.

Given this premise, we assume that measurement errors are normally distributed with a mean of zero and a known standard deviation which is a percent of the measured value. We simulate a large number of observations (with random measurement errors) for each functional form of Q_z. We hypothesize that the variation in Q_z is normally distributed with a mean of zero. The size of the standard deviation of Q_z will depend on both the Q_z-form and on the magnitude of the measurement errors.

We present results of the simulation for three functional forms of Q_z. The size of the measurement error will be varied to determine the effect, if any, on the distribution of Q_z.

DESCRIPTION OF SIMULATION MODEL

The simulation model assumes a complete representation of Q_z. That is, a representation for which the similarity measure would equal zero if no measurement error exists. Any variation we observe in Q_z will, thus, be due to transmission of measurement errors of the parameters.

Newton's law, for example, has as one complete representation, $Q_z=s/vt$. We will describe the simulation approach for this example. We similarly treat the additional forms of Q_z to be investigated. The user selects specific values for a, v and t; s is calculated from Newton's Law ($s=0.5at^2+vt$). Likewise the true value for Q_z is defined by s, v and t ($Q_z=s/vt$). A set of measurements are simulated for these values of s, v and t by generating and adding normally distributed independent random errors. These instances belong to one similarity class. The user can assign the standard deviation of the error as a fraction of the measured value.

The exact value of Q_z is known. A "real" value of Q_z is calculated after the measurement errors are introduced into the data. Then the difference between the exact and the actual (with errors) value of Q_z is determined. The process is repeated for a large number of times (n=100 in our examples below). The mean and variance of Q_z (that is, the function f above) are calculated for these 100 instances. Note, the similarity measure is the sum of absolute differences between the mean of Q_z and each instance divided by the number of instances. The statistical distribution of Q_z is tested for fit to a Normal distribution using the chi-squared test.

EXPERIMENTAL DESIGN

We are analyzing the effect of three factors. Both the size of measurement error and the parameter with error may affect the resulting error in the similarity measure. The third factor is the functional form of the relationship between Q_z and the measured variables. In our case, the functional form is fixed for a particular situation. It is, one might say, defined by the dimensional analysis being pursued. Changing the size of measurement error within one functional form will provide a complete analysis for this particular functional form.

Thus our experimental design reduces to a two factor design with replications for the different functional forms of interest. A typical two-factor experiment measures the effect of two factors, say pressure and temperature, on yield. In such a case, the recommended procedure is to change the factors together and not one at a time. The "together" approach follows the response surface and avoids false conclusions which might otherwise result. We follow this approach in our experimental plan.

Each row in Table 1 summarizes twenty-five simulation experiments for each of three functional forms. The first five rows monitor the effect of increasing measurement error in all variables simultaneously. The next twelve measure the effect of one variable having much larger error that the others. The chi-squared test for normal divides the 100 observations from each simulation into six cells. Expected frequencies are calculated using sample mean and sample standard as estimates for population parameters. The calculated chi-squared test statistic is compared to a rejection value for alpha = 0.01 and three degrees of freedom (critical value = 11.343).

ANALYSIS OF SIMULATION RESULTS

We found strong support for the assumption that the errors in Q_z are normally distributed if the measurement errors are small. For measurement errors with standard deviation of less than 5 percent, the distribution of errors easily passes the chi-squared test. If the errors are larger than 10 percent and if the variable appears only in the numerator of Q_z, the normality assumption continues to pass the chi-squared goodness-of-fit test. However, for errors of more than 10 percent for a term appearing in the denominator of Q_z, the normal assumption is rejected far more frequently than would occur by chance. The results are even more non-normal if the variable is raised to a power in the denominator.

The magnitude of the error in Q_z is quite consistent if errors are small. For errors of less than 5 percent, the error in Q_z is between 1.7 and 2.5 times the original measurement error. The larger error transmission occurs when a variable is raised to a power in the denominator.

Measurement Error: standard deviation of errors, percent of mean value				Variability in Q_z: standard deviation of Q_z, percent of mean			Variability in Q_z: number of 25 failing chi-squared test, $\alpha = 0.01$		
s	v	t	a	s/vt	sa/v^2	s/at^2	s/vt	sa/v^2	s/at^2
.01	.01	.01	.01	.017	.025	.025	0	0	0
.1	.1	.1	.1	.17	.25	.24	0	0	0
1.0	1.0	1.0	1.0	1.7	2.5	2.4	0	0	0
5.0	5.0	5.0	5.0	8.7	12.4	12.6	1	1	0
10.0	10.0	10.0	10.0	17.4	25.7	25.8	1	4	10
1.0	1.0	5.0	1.0	5.2	2.4	10.0	0	1	0
1.0	1.0	10.0	1.0	10.3	2.5	21.6	0	0	5
1.0	1.0	15.0	1.0	16.1	2.4	35.0	6	2	19
1.0	1.0	20.0	1.0	22.0	2.5	53.0	14	0	23
10.0	1.0	1.0	1.0	10.2	10.4	10.4	1	0	0
30.0	1.0	1.0	1.0	30.4	29.8	29.2	0	0	0
90.0	1.0	1.0	1.0	87.1	89.0	92.6	0	1	0
1.0	5.0	1.0	1.0	5.1	10.1	2.5	0	0	1
1.0	10.0	1.0	1.0	10.3	21.5	2.4	3	4	0
1.0	1.0	1.0	10.0	1.7	10.3	10.3	0	0	3
1.0	1.0	1.0	15.0	1.7	15.2	16.3	0	1	6
1.0	1.0	1.0	20.0	1.7	20.2	22.9	0	0	12

Table 1
Results of Simulation Runs
Sample Size of 100 for Each Simulation
25 Simulations for Each Functional Form of Q_z

CONCLUSIONS

Analysis of the COPER similarity measure using the probabilistic/statistical approach is helpful but in a limited way. One must assume that measurement errors (noise in the input data) are fairly small (as a percent of measured value) and that the errors are themselves normally distributed. In such a case, the uncertainty in Q_z resulting from measurement error is normally distributed. In addition, given these same assumptions, the standard deviation of this transmitted measurement error is between 1.7 and 2.5 times the original measurement error when defined as a percent of the "true" value. The 1.7 multiplier applies when only linear functions of the parameters appear in the denominator of Q_z. The 2.5 multiplier applies when the squared value of a parameter appears in the denominator of Q_z.

We plan to incorporate these results into COPER. Given information about the size of the measurement errors, the system can decide if the error in Q_z is well defined. That is, if the situation fits the small error conditions described in the simulation analysis above. The system can further decide, based on the functional form of Q_z, the approximate size and distribution of errors in Q_z. Thus, under the limited conditions defined above, COPER will incorporate analysis of uncertainty caused by noise in the input data.

However, a full resolution has not been attained. We plan to evaluate alternative approaches such as fuzzy set theory and Dempster-Shafer belief functions for dealing with uncertainty in our attempts to expand the decision rules for dealing with uncertainty within the COPER system.

REFERENCES

Birkhoff, G., (1960). *Hydrodynamics. A study in logic, fact and similitude.*, Princeton University Press, Princeton.

Drobot, S. (1953). On the foundations of dimensional analysis. *Studia Mathematica*, 14, pp.84-89.

Kokar, M., M. (1985). *On invariance in dimensional analysis* (Technical Report MMK-2/85). Boston, MA: Northeastern University, College of Engineering.

Kokar, M., M., (1986a). Determining Arguments of Invariant Functional Descriptions, *Machine Learning*, 1, pp.403-422.

Kokar, M., M., (1986b). Discovering functional formulas through changing representation base. *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, pp.455-459.

Lewis E.E., (1987), *Introduction to Reliability Engineering*, Wiley, New York, N.Y., p. 61.

Mendenhall, W., Scheaffer, R., L. and Wackerly, D., D. (1986), *Mathematical Statistics with Applications*, PWS Publishers, Boston, MA, p. 253.

Whitney, H., (1968), The Mathematics of Physical Quantities, part I and II, *American Mathematical Monthly*, pp. 115-138 and 227-256.

APPENDIX A

Consider the variance of $(1/Z)$ where Z is a normal random variable with mean = 0 and standard deviation = 1. Note, any normal random variable X, with mean = μ and standard deviation = σ, can be transformed to a standard normal Z using the relationship:

$$Z=(X-\mu)/\sigma.$$

Variance $1/Z$

$$= E\{[1/Z-E(1/Z)]^2\} = E\{1/Z^2-2(1/Z)(E(1/Z)+[E(1/Z)]^2\}$$
$$= E(1/Z^2)-[E(1/Z)]^2$$

Consider $E(1/Z^2)$

$$E(1/Z^2) = \int_{-\infty}^{+\infty} 1/z^2 f(z)\,dz = \int_{-\infty}^{+\infty} 1/z^2\, 1/2e^{-z^2}\,dz$$

Consider the interval from -1 to +1, for this interval $f(z)\,(=1/2e^{-z^2})$ has an upper bound of +1/2 and a lower bound of $1/2e^{-1}$. Let $K = 1/2e^{-1}$.

Then

$$\int_{-1}^{+1} (1/z^2)\, 1/2\, e^{-z^2}\,dz \geq \int_{-1}^{+1} K\,(1/z^2)dz = K\int_{-1}^{+1} 1/z^2 dz$$

But this integral has a value of infinity, thus $E(1/Z^2)$ is greater than or equal to infinity. And therefore, the variance of $(1/Z)$ can not be found analytically.

KNOWLEDGE-BASED PROJECT MANAGEMENT: WORK EFFORT ESTIMATION

Vijay Kanabar
University of Winnipeg

ABSTRACT

Planning and estimating work effort for a project is one of the most difficult activities in Project Management. It is also a critical activity since preliminary estimates translate directly to estimated costs. Projects with lower estimated costs end up with insufficient funding, while projects with high estimates are simply not considered for development. To assist with estimation a strategy integrating knowledge-based techniques with procedural techniques is proposed. An Integrated Planning Model (IPM) based on this concept is described in this paper.

1.0 Introduction to Project Management

Managing a project using the critical path method typically involves three main stages- Planning, Scheduling and Control.
During the Planning stage the project is broken down into smaller more manageable components called activities. Work effort is estimated for each of these activities using past experience as a guide. Formulas or Models, and actual historical data are also used.

Once the work effort is completed a network is created to show the sequence of activities that make up the entire project. Several project management tools are available for the above purpose. They range from powerful mainframe based products such as IBM's Application System[1] to relatively smaller project management tools based on microcomputers.

The next stage is Scheduling - here we map the activities to a calendar, and determine start and finish dates for each task.

The last stage is Control, and this ensures that the entire project is completed on time and within budget. Good control also ensures that the end products are of good quality.

2.0 Automated Approach

While several software tools are available for project management, few provide assistance with estimating. Existing tools assist project developers only after activities have been defined and the work effort estimated. Subsequently, useful networks are drawn, the Critical Path traced, reports generated, and graphs such as GANTT drawn.

At the outset it would appear that it is meaningless for any tool to support work effort estimation; after all, every project is different from another! But on further analysis, it is evident that this logic is incorrect. As explained earlier, projects are always broken down finely into tasks and activities. Any new project will therefore have some activity that was performed earlier (e.g., all projects involve creating a users manual). It is then possible to borrow such estimates for the new project. To quote Meilir Page-Jones, "the best way to assign a cost to a given task is to identify the cost for an identical task performed earlier in the shop." [2]

Unfortunately few project managers have the opportunity to do so, as no useful data about previous attempts ever get recorded. Several explanations are offered for this lack of data by Meilir, including "My people have no time to collect project data", and "Nobody in this shop has the statistical skills to apply the collected data meaningfully."

2.1 Metrics Group

To get around this problem some Data Processing shops have established a Metrics Group.[3] The members of this group are specialists in measurement and estimating and they acquire their skills over many projects. They also function independently off the project manager and therefore are not subject to political pressure and bias.

There are several advantages with this approach:
a) Members acquire specialised estimation skills.

b) They will have acquired adequate statistical skills.
c) Project Managers and other developers can rely on this group to obtain better results.

But there are also several disadvantages:
a) Maintaining such an exclusive group can be expensive.
b) Benefits will be seen only after the group is well established.
c) High staff turnover in this group can be devastating.
d) Dividing authority between the Metrics group and the Project Manager can be tricky.

2.2 Knowledge-Based Systems

An alternative strategy would be to develop knowledge-based systems to assist with planning and estimation. Such systems provide the following advantages, as stated *in Waterman* [4].

a) Permanent resource available.
b) Expertise is easy to transfer.
c) Consistent procedures used.
d) Affordable.

Accordingly, a new *Integrated Planning Model* (IPM) is proposed.
The model uses knowledge-based, procedural and statistical techniques, and can integrate with existing software systems (e.g., project management tools, database systems, spreadsheets and other estimation models).

3.0 IPM Design Overview

The architecture of IPM is illustrated in Figure 2. The heart of the system is the "IPM Kernel," which determines the recommended work effort.

The Knowledge-base component is used to store facts and rules. (Knowledge-based systems are made up of a knowledge-base and an inference engine). Up-to-date information for a given domain is stored in the knowledge-base. The inference engine is responsible for processing the information in the knowledge-base and coming up with solutions. (See Figure 1 for more details). Two expert

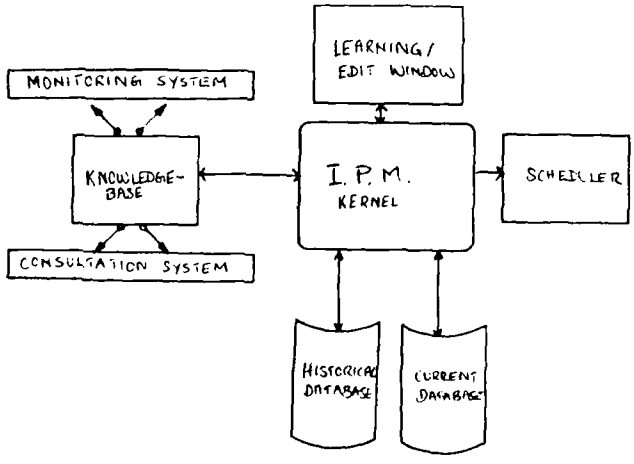


FIG 2. IPM DESIGN OVERVIEW

systems reside here - the Monitoring System and the Consultation System.

The Historical Database contains actual data from previous projects (e.g., type of project, time taken to complete, activities, resources utilized). The Current Database stores information about existing projects

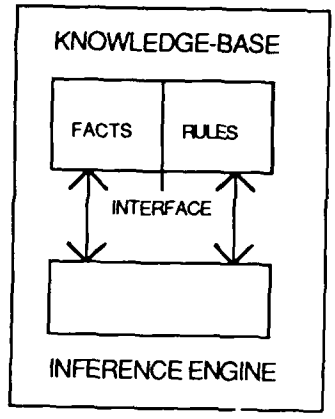


FIGURE 1. STRUCTURE OF A KNOWLEDGE-BASED SYSTEM

(projects just initiated and projects not yet completed.) On completion of a project the actual hours are transferred to the Historical Database.

The Learning module updates the knowledge-base with current information. It also acts as an edit window to remove outdated information, or delete incorrect files (that is entire projects can be removed).

Finally, the Scheduler is responsible for receiving the recommended estimates, and generating reports and charts, such as the Project Schedule, Project Calendar, Resource Table, Critical Path.

3.1 IPM Execution

Execution of IPM involves the following phases:

1. Identifying the project as belonging to a particular category.

This decision is made manually by the project manger on the basis of the Planning Document, Requirements Analysis and estimated number of *thousands of Delivered Source Instructions* (DSI). For instance a popular model such as Boehm's COCOMO can be used[5]. He considers the development of a software system for a company that has determined that their program will have roughly 32,000 DSI. The following equations of the COCOMO model are used to estimate important characteristics of such a software system.

Effort: MM = 2.4 (32)exp 1.05 = 91 man-months
(One man-month = 152 hours of working time)

Schedule: Estimated Development = 2.5 (91)exp 0.38 = 14 months

Average Staffing: 91 man-months/14 months = 6.5 Personnel

On the basis of such estimates, classify the project as belonging to either a) Small Intermediate b) Intermediate c) Large.

Table I:

Subphase 1.3 User Requirements	SRWE (Hours) a.	b.	c.	d.	e.	Min	Max	Ave	
1.3.1 Set Up The Project	13	12	15	10	12	10	15	12	
1.3.2 Review Existing System	15	2	19	25	12	2	25	15	*
1.3.3 Interview Users	3	4	4	3	2	2	4	3	
1.3.4 Document User Requirements	14	12	15	11	13	11	15	13	
1.3.5 Review Document with Users	0	0	0	0	0	0	0	0	

2. Generation of a Work Break Down Structure.

The next step involves generation of the Work Break Down Structure (WBS). This structure is a hierarchy that identifies all the end products. All project and development work are defined here as activities. IPM asks some more questions about the nature, scope and size of the project and automatically provides a default WBS. There is an unique WBS for each of the following categories of projects- Small, Intermediate and Large. (A very reliable WBS can be generated if a related project can be identified in the Historic Database). If the WBS supplies by IPM is not completely satisfactory it has to be modified. This customization is important as every project is slightly different from another.

Sample WBS generated for a small project is shown below:

Subphase 1.3 User Requirements

1.3.1	Set Up The Project
1.3.2	Review Existing System
1.3.3	Interview Users
1.3.4	Document User Requirements
1.3.5	Review Document with Users

3. Loading the WBS with Raw Work Effort.

The above WBS is now ready for loading with raw work effort. IPM supplies a Suggested Raw Work Effort (SRWE) for each of the above activities. This is the simple mean of the previous estimates for a comparable project. If data for any particular activity does not exist in the database the SRWE field is simply left blank. See activity 1.3.5 for an example.

Subphase 1.3 User Requirements		SRWE (Hours)
1.3.1	Set Up The Project	12
1.3.2	Review Existing System	15*
1.3.3	Interview Users	3
1.3.4	Document User Requirements	13
1.3.5	Review Document with Users	

As indicated above SRWE numbers are simple statistical averages.
If the sample size for SRWE is less than 5, or if the variance is large, the SRWE value will be flagged with an asterisk, (such as in 1.3.2.) Alternatively, three columns can be displayed, the Minimum SRWE, Average SRWE and Maximum SRWE as illustrated in Table I.

4. Loading Raw Work Effort

If the SRWE values appear to be unsatisfactory they may changed completely. This may come about after a consultation with the Knowledge-base system. Also at this stage activities with null SRWE's are given an estimated value (manually by the Project Manager). For example, 1.3.5 is given a value of 2 hours.

Subphase 1.3 User Requirements		SRWE RWE (Hours)
1.3.1	Set Up The Project	12
1.3.2	Review Existing System	15 *
1.3.3	Interview Users	3
1.3.4	Document User Requirements	13
1.3.5	Review Document with Users	2

Assumptions are noted down in the database as to why a particular RWE value was given. This will come in handy if the estimates have to be revised again in the future.

5. Invoking the Monitoring System:

The Monitoring System (which is an expert system) evaluates the estimated hours for the entire project. It can also make selective analysis of the different phases and subphases. A comment such as this might occur after the initial evaluation:

" The total time allocated for System Testing is too low. The following are historic maximum, minimum and average values ...
The following rule of thumb for scheduling a software task is recommended [Brooks] [6]: 1/3 Planning, 1/6 Coding, 1/4 Component Test and Early System Test and 1/4 System Testing. You want to revise the estimated figures for Phase 4."

Another general purpose comment such as this might also occur:
"Application Prototyping is strongly recommended for small business applications. This has to be done in Phase 2."

6. Loading Task Performers (Invoking the Consultation System):

The task performers have to be assigned at this stage. The RWE values are divided and adjusted according to their individual skills. The Consultation system (expert system) is invoked at this stage. Unlike the Monitoring System which can be classified as a general purpose system, the Consultation System is a specialised knowledge-base based on the local environment (i.e., a particular Data Processing shop. Consequently, each organization has to have its own consultation system. This system best functions in the interactive mode. The following queries can be asked.

"List the names of all CICS programmers."

"If John Smith were to do the coding, how long would it take him?"

"What is the D.P. shop policy on Documentation?"

Information on individual software packages can also be queried. For instance the following information may be helpful,

"FASTCODE is a good package for Report Writing, it can generate a simple report in 10 minutes; it has a good training manual and takes approximately 8 hours to complete the tutorial".

It is obvious that the Consultation System can act as a good training tool as well. Valuable information can be extracted from the knowledge-base.

It is evident from the above examples that the Consultation Database is dependent on the local environment. Local technical experience with knowledge engineering and designing expert systems must be available, otherwise, this component will not be useful. However, this phase can be completed manually (as is commonly done).

7. Scheduler

At this stage the estimates are passed to the Scheduler. Several project management tools assist with scheduling. Based on the activity network completion dates for all the activities are alloted.

8. Iteration

Needless to state, several iterations of the above will occur during the life of a project. The first estimates will be revised repeatedly as the project progresses.

3.2 Learning in IPM

The term "learn" is used here in the sense that the IPM knowledge-base expands to accommodate additional data and information. The following examples serve to illustrate the point.

3.2.1 Adding Structured Information

Information added to the knowledge-base can be structured or unstructured. When the information is structured it is possible to use the information during estimation.

For example, if a project manager acquires information about a new "faster and user-friendly" software package for application prototyping, it would be helpful if the knowledge-base had access to this new information. This can be implemented as follows - consider this dialog

Do you want to add new information to the knowledge base?
>>Yes

To what database must this new information be added to?
>>Software Packages

Please enter the following information

Name of package: 6GL
Price: 650.00
Average Time taken to complete tutorial: 4 Hours
Estimated Productivity Factor (out of 10): 7
Additional Information about the software:

When all the questions have been answered the above information is inserted into the database and it will be considered when a new schedule is generated.

3.2.2 Adding Unstructured Information

Unstructured information can also be added to the knowledge-base.
This is simply a note-pad or a memo field. Consider the following comments about the Prog Shop:

(On invoking the Learning Module)
Do you have any comments about this Activity?
>> I have experienced considerable delay in obtaining printed forms from the Print Shop. This has considerably delayed the development process. It would help future projects if printing requisitioning is done in the preceding phase.

Such comments can be quite useful to a novice Project Manager reviewing historical files.

4.0 Conclusion

The automated approach to planning and estimating will provide benefits only if the following criteria are satisfied:

1. Projects are consistent in size and scope.

2. Actual historical data are stored in the Historical Database (ie., the completed estimates not the initial estimates.)

3. Sincere attempt is made to document actual experience (such as the preceeding example about Print Shoop delays) into the Monitoring and Consultation systems.)

Finally, it appears that such an automated system can also be a front end module in the Computer Assisted Software Engineering (CASE) system architecture. Integration of management and development tools is one of the main goals of the CASE system architecture.

1. IBM, "Managing Projects with Application System", Release 4, Product Number 5767-001, 1986

2. Meilir Page-Jones, "Practical Project Management", p 14, Dorset House Publishing Co., New York, 1985.

3.De Marco, t., "Structured Analysis and System Specification", Yourdon Press, New York, 1978

4. Waterman, Donald, "A Guide to Expert Systems", p 12, Addison-Wesley, 1986

5. Boehm, Barry, "Software Engineering Economics", Prentice Hall, 1981

6. Brooks, Fredrick, "The Mythical Man-Month", Addison Wesley, 1984

COMBINING KNOWLEDGE ACQUISITION AND CLASSICAL STATISTICAL TECHNIQUES IN THE DEVELOPMENT OF A VETERINARY MEDICAL EXPERT SYSTEM

by

Mary McLeish[+], Matthew Cecile
Department of Computing and Information Science
University of Guelph

Larry Rendell
Dept. of Computer Science
University of Illinois

Dr. P. Pascoe
OVC
Guelph, Ontario

Abstract

This paper explores a number of different tools for aiding the medical diagnostic process in the domain of equine colic. Methodologies from the machine learning and knowledge acquisition world are compared to more classical statistical techniques and reasons are given why these methods can *complement and enhance* the diagnosis. A variation on a weight of evidence formula due to I. Good, which uses the notion of the probability of a fuzzy event, is introduced and some initial results are presented.

1. Introduction

This paper discusses the progress to date on a large project undertaken primarily at the University of Guelph, which is the home to one of Canada's major (and few) Veterinary Colleges. The larger goal of the project is to build a general diagnostic shell for veterinary medicine. Work has begun on a *prototype involving* the diagnosis of surgical versus medical colic in *horses*. This is a significant problem in veterinary medicine having led to many studies, use of diagnostic charts etc. to aid owners and veterinarians in recognizing serious cases [6,7,24]. Horses suspected of requiring surgery must be shipped at a significant cost to the veterinary hospital, where further tests are conducted and a final decision is made. We are endeavoring to provide a computerized diagnostic *tool for use in the hospital* as well as on a remote access basis for practicing veterinarians.

The animal hospital at Guelph has a computer system called VMIMS (Veterinary Medical Information Management System). This system was originally programmed in Sharp APL and handles usual admission and billing procedures. However, the system also stores a considerable amount of medical information on each patient including bacteriology, clinical pathology, parasitology, radiology and other patient information such as age, sex, breed, presenting complaint, treatment procedures, [illegible]. In clinical pathology, [illegible] automatically generated by the lab equipment. The database currently holds about 18,000 [illegible] 30,000 [illegible] occupying 500 megabytes [illegible] storage. The current project is being designed to make use of the vast amounts of archive data [illegible] the diagnostic process. Other database issues concerning the implementation of our system are discussed in [illegible] and [illegible] in [12]. [illegible] of the proposed system is given in [3].

Medical expert systems have been under development for [illegible] years, especially [illegible] seen in major projects like MYCIN (Stanford) [3]. These recent systems have been largely based on the assumption that to have expert capability, they must somehow mimic the behavior of experts. Earlier work, using mathematical formalisms (decision analysis, pattern matching, etc.) were largely discarded and attention to the study of the actual problem-solving behavior of experienced clinicians. *In a recent paper [21] by Drs.* Patil, Szolovits and Schwartz, it is suggested that the time has come to *link the old* with the new: "now that much of the AI community has turned to causal, pathophysiologic reasoning, it has become apparent that some of the earlier, discarded strategies may have important value in enhancing the performance of new programs..." The authors recognize the difficulty of this approach when they state that "an extensive research effort is required before all these techniques can be *incorporated* into a single program."

We are experimenting *with the use of statistical* techniques and at the same time developing a rule-based system from expert opinions. This paper is primarily concerned with finding the right tools to analyze the data before comparing such results with the "expert" rules component. Section 2.1 discusses the use of discriminant analysis and logistic regression. Section 2.2 looks at a method of Bayesian classification and section 2.3 considers an inductive learning technique due to the 3rd author and *another method*, due to R. Quinlan [18] also arising from the machine learning community. Section 3 introduces a method *using fuzzy* sets which also incorporates a weight of evidence formula due to I. Good [8,9]. In Section 3.4, an example is given and the effectiveness of the various methods is compared. Finally section 4 highlights work which is planned and is currently in progress.

2.1 Classical Statistical Methods

The data used for these studies represented 253 horses present at the teaching hospital at Guelph. The horses were all subjected to the same clinical tests and the same pathology data were collected. This data set was used for all the studies discussed in this paper. Outcome information of the following type was available: whether surgery was performed, whether or not a surgical lesion was actually found and the final state of the animal (i.e. lived, died, euthanized). The objective of the study was to see which variables changed during examination of the horses with abdominal pain were significant in differentiating between horses that required surgery versus medical treatment. The [illegible] parameters [illegible] were [illegible] and [illegible]

temperature, heart rate, respiratory rate, temperature of extremities, colour of mucus membranes, capillary refill times, presence and severity of abdominal pain, abdominal distension, peristalsis, and the results of naso-gastric intubation and rectal examination. Clinico-pathological parameters evaluated included hematocrit (HCT) and the total plasma concentration of abdominal fluid. These variables were sometimes continuous and when descriptive (pain levels, etc.) were translated into discrete integer variables. Missing data was handled by elimination of cases. There are 20 parameters in each of the two data sets (*clin* and *clin-path*).

A multiple stepwise discriminant analysis in a recursive partition model was used to determine a decision protocol. The decision protocol was validated by a jackknife classification and also by evaluation with referral population in which the prevalence of surgical patients was 61% c.f. 6, 7 . The significant parameters were found to be abdominal pain, distension and to a lesser extent, the color of abdominal fluid. The use of the decision tree yielded a significant number of false positives and virtually eliminated false negatives in one study. Unnecessary surgery is even more undesirable in animals than humans due to costs (usually borne by the owner) and the debilitating effects of surgery on a productive animal. Other difficulties with these results concerned the fact that the clinical pathology data appeared entirely non-predictive - a result contrary to the medical belief that, at least in serious cases, certain of these measured parameters do change significantly. Discriminant analysis can miss effects when variables are not linearly behaved. Missing data was another serious problem. Other methods described in section 3 helped overcome some of these problems.

Logistic Regression [1] was also run on the same data set. Here, a regression model of the form

$$Y_i = \log_e \frac{P_i}{1-P_i} = \beta_0 + \beta_1 X_1 + \cdots \beta_k X_k$$

is used where the β_i's are slope parameters relating each of the X_i independent variables to the Y_i's (log odd's ratio). Appropriate transformations were made to account for nominal data. The data was run for all three possibilities: surgical lesion found (SL), surgery performed (S) and outcome (O). The outcome, O, can take on the values of lived, died or euthanized. Pulse, distension and a variable representing the presence of firm feces in the large intestine (A2) were the significant predictors. However, testing against whether the doctors decided to do surgery, pain and A2 were the most significant. (These results were obtained from the clinical data only.) Outcome found several other variables to be significant: the probability of death is increased by pain, cold extremities, a high packed cell volume and low NGG reading (Naso-gastric tube emissions). Again problems were caused by the presence of missing data. Some modifications of these results were found when missing data were estimated using stepwise regression methods. The reasons for the missing data were complicated and various - sometimes records ended because euthanasia was chosen as a solution for cost reasons. Thus estimating missing data was not a very reliable technique.

Although one can change the levels of significance (95% was used), the regression models assume a model of behaviour of the variables (here, linear). Another problem of the results of the two analyzes presented concerns the reliance on the measurements of pain and type of abdominal distension. These variables are extremely subjective and carry a very large measure of error. Much other medical data is collected on the cases of a more precise nature. Some of the other methods to be discussed make more use of these other measurements in formulating their diagnosis.

2.2 Bayesian Classification

This methodology uses Baye's Theorem to discover an optimal set of classes for a given set of examples. These classes can be used to make predictions or give insight into patterns that occur in a particular domain. Unlike many clustering techniques, there is no need to specify a "similarity" or "distance" measure or the number of classes in advance. The approach here finds the most probable classification given the data. It allows for both category-valued information and real-valued information. For further details on the theory, the reader is referred to [4,5,22].

A program called Autoclass I (see also [5]) was run on the combined clinical and pathological data sets. All 51 variables were included, that is, all outcome possibilities, as described in section 2, were included as variables and lesion type (four possibilities) was also added. A total of 13 classes were found and in most cases the probabilities of a horse belonging to a class were 1.00. The type of information available consisted of relative influence values for every attribute to the over-all classification. For each class and each attribute, influence values are produced indicating the relative influence of each attribute to that class. This information is available in tabular and graphical form.

The classes provide related groups of cases which are useful for case studies (OVC is also a teaching institution). The information may be used predictively. For example, the class with the highest normalized weight, class 0, was found to have surgical lesion as a very high influence factor. The variables of abdominal distension, pulse, abdomen (containing A2 mentioned earlier) and pain, found significant by earlier methods, were also influential factors for this class. Some other variables not flagged by earlier methods were found to be influential as well (total protein levels and abdominocentesis, in particular). The horses in class 0 were found to not have surgical lesions. It is thus possible to see from the features of horses in this class which attributes and what type of attribute values are significant for this to be the case. New cases can be categorized into classes according to their attribute values and if found to be in class 0, this would indicate a very small chance of surgery being required. Class 1, however, is predominantly a class where surgeries are required.

Actually, there is a wealth of information to be gleaned from the results and much of this interpretive work is ongoing at this time. One may infer that certain variables are not very predictive. For example, a variable distinguishing young from old animals has low influence value in all major classes, but is slightly more significant in a couple of the smallest classes. Studying these small classes, however, can be particularly fascinat-

ing because they flag situations which are more unusual. That is, methods which simply find variables which are *usually* the most predictive, cannot perform well on cases which do not conform to the normal pattern. Other classes pinpoint cases difficult to diagnose. Class 12 contains three cases which were all operated on, but only one out of three was actually found to have a surgical lesion. Two cases had a simple large colon obturation and the third a large colon volvulus or torsion (requiring surgery). That is, two unnecessary surgeries were performed. A close study of the influential variables and their parameter values for this class of cases - with very close but importantly different diagnoses - provides extremely valuable information.

A significant question arises as to whether or not outcome information should be included initially in the program run. If the data were highly predictive, it should not matter. However, it was not clear from the onset how true this was. The data was run twice again: once with all outcome information removed and once with the doctor's decision information deleted. When all outcome information was removed, one notes that some interesting prognosis information still remains. Class 1 indicated cases virtually all of which lived and whose condition, whether or not surgery was required, was generally good. The question often arises whether or not to operate even given that the animal has a lesion if the general prognosis is bad. This class would indicate that surgery should be performed in such cases. Class 2 (42 cases) was extremely well discriminated with 91% having a surgical lesion. Of the remaining classes (again there were 13 in total), 3 4 were also reasonably well discriminated on the basis of lesion. Others flagged items such as young animals or cases that had very poor over-all prognosis.

New data has been obtained and code written to take new cases and determine a probability distribution, for this case over the classes, from which a probability of outcome may be calculated. It is interesting to note that the use of Bayesian classification for medical diagnosis in this fashion is in a sense a mathematical model of the mental process the clinicians themselves use. That is, they try to think of similar cases and what happened to those cases in making predictions. The technique for making predictions outlined above provides a sophisticated automation of this process. It is impossible within the scope of this paper to document all the information obtained from using Bayesian inductive inference. A sample case will be provided in Section 3 4 which will show the usefulness of combining information from the results of Bayesian inductive inference with the other methodologies.

2.3 Other Induction Techniques

The probabilistic learning system developed by the third author (cf. 19, 20) also classifies data. Classes are optimally discriminated according to an inductive criterion which is essentially information-theoretic. To accommodate dynamic and uncertain learning, PLS1 represents concepts both as prototypes and as hypertetrangles. To accommodate uncertainty and noise, the inductive criterion incorporates both probability and its error. To facilitate dynamic learning, classes of probability are clustered, updated and refined. The references describe the method here in detail. We give here one of the results of the application of the preliminary data.

Two classes of data were given to the program (**1**: no surgery, **2**:surgery) and 14 clinical pathology variables were processed according to the PLS1 algorithm. The results can be interpreted as rules and also flag the most prominent variables. The current version of PLS1 required that the data be scaled between 0 and 255. The variable X1 represents total cell numbers, X8 is mesothelial cells and X13 is inflammation. These were found to be the most significant variables for the prediction of no surgery. For the purpose of predicting surgery, again X1 and X13 were significant, as well as X9, a measure of degenerate cells. Uncertain rules can also be obtained from the results. Further work is being done to revise the learning algorithm to be able to handle missing data without losing entire cases (i.e. all parameter values when only one is absent). The statistical methods lose all case information and it would be a considerable advantage to employ a method less sensitive to this problem.

This technique produces diagnostic rules of the form:

Class 1.

$[\ 1 \leq x1 \leq 1\ ]\ [\ 0 \leq x8 \leq 26\ ]$
$[\ 0 \leq x13 \leq 0\ ]$, p with a utility value of .8333

Class 2.

$[\ 2 \leq x1 \leq 255\ ]\ [\ 1 \leq x9 \leq 255\ ]$
$[\ 1 \leq x13 \leq 255\ ]$, with a utility value of .8462.

Actually, for each class there are several rules with different utility functions. A very low utility function value would indicate more probable membership in the other class. For the example given, not abnormal values of x1 and x8 combined with no inflammation is indicative of a medical colic at level .83, whereas a higher cell count (more abnormal), together with some inflammation and the presence of degenerate cells indicates a surgical case at a level of .85. Keeping 50 cases aside for test purposes, the performance of this system for outcome (lived, died or euthanized) was 74% correct diagnoses. For the prediction of lesion -no lesion the result was 64%. However, the initial training set was not particularly large when missing data was taken into account and we have now just completed the collection of new compatible data which is being tested. The initial performance for outcome was better than the doctors predictions and for lesion -no lesion was lower (by about 8%).

Quinlan's algorithm 18 for rule extraction has been run on the clinical data. Continuous data has been converted to discrete values. The important variables are mucus membranes, peristalsis, rectal temperature, packed cell volume, pulse, abdominal condition, and nasogastric reflux. A number of variables appeared significant using this technique which were deemed unimportant using discriminant analysis. A decision tree was generated but some implementational difficulties due to the number of variables used and missing data led to a tree which was not very complete or reliable. These problems are currently being addressed to develop a more robust version of the algorithm. The method in the following section uses all variables provided rather than reducing the variable set and bases the diagnosis on only a few parameters.

3.1 Overview

This section describes a method of evidence combination which performs Bayesian updating using evidence that may be best modelled using an infinite valued logic such as that which fuzzy set theory provides. The methodology described in this section provides a unified approach for intelligent reasoning in domains that include probabilistic uncertainty as well as interpretive or "fuzzy" uncertainty.

A formulation central to several components of the methodology is that of the "weight of evidence" and is therefore introduced in section 3.2. The description (and justification) of the use of an infinite valued logic is presented in section 3.3 and explains how "important" symptom sets are used. Section 3.4 relates the performance of this method in the domain of equine colic diagnoses.

3.2 The Weight of Evidence

A.M. Turing originally developed a formulation for what he called the "weight of evidence provided by the evidence E towards the hypothesis H" or W(H:E). Good [8,9] has subsequently investigated many of the properties and uses of Turing's formulation which is expressed as:

$$W(H:E) = \log\left\{\frac{p(E|H)}{p(E|\overline{H})}\right\} \quad Or \quad W(H:E) = \log\left\{\frac{O(H|E)}{O(H)}\right\}$$

where O(H) represents the odds of H, $\frac{p(H)}{p(\overline{H})}$. Weight of evidence plays the following part in Bayesian inference:

Prior log odds + $\sum_i$ *weight of evidence*$_i$ = posterior log odds

A weight of evidence which is highly negative implies that there is significant reason to believe in $\overline{H}$ while a positive W(H:E) supports H. This formulation has been most notably used in a decision support system called GLADYS developed by Spiegelhalter [23].

In any formulation for evidence combination using higher order joint probabilities there exists the problem of evidence that may appear in many different ways. For example, a patient has the following important symptom groups: (High pain), (High pain, high temp.), (*High pain, high temp, high pulse*). *Which of these symptom groups should be used?* Using more than one would obviously be counting the evidence a number of times. The rule we have chosen to resolve this situation is to choose the symptom group based on a combination of the group's size, weight, and error. In this way we may balance these factors depending upon their importance in the domain. *For example, if higher order dependency* is not evident in a domain then the size of a group is of little importance.

3.3 Events as Strong α - Level Subsets

Infinite valued logic (IVL) is based on the belief that logical propositions are not necessarily just true or false but may fall anywhere in [0,1]. Fuzzy set theory is one common IVL which provides a means of representing the *truth of a subjective or interpretive statement*. For example, what a physician considers to be a "normal" temperature may be unsure or "fuzzy" for certain values. For these values the physician may say that the temperature is "sort of normal" or "sort of normal but also sort of high". This is a different and separate concept from the probability of an event (using either the belief or likelihood interpretations). Implicitly, probability theory (in both interpretations) assume that an event either happens or does not (is true or false). On the practical side, we have found that the concept and estimation of membership functions is intuitively easy for physicians.

Let F be a **fuzzy subset** of a universe, U. F is a set of pairs $\{x, \mu_F(x), x \in U\}$ where $\mu_F(x)$ takes a value in [0,1]. This value is called the grade of membership of x in F and is a measure of the level of truth of the statement "x is a member of the set F".

A strong **α - level subset**, A_α, of F is a fuzzy set whose elements must have a grade of membership of $\geq \alpha$. Formally defined,

$$A_\alpha = \{\mu_{A_\alpha}(x) \mid \mu_{A_\alpha}(x) \geq \alpha\}$$

For example, if we have the fuzzy set F = {x1/0.2, x2/0.7, x3/0.0, x4/0.4} then the strong α - level set $A_{\alpha=0.2}$ = {x1/0.2, x2/0.7, x4/0.4}.

Because we wish to perform probabilistic inference we need to have a means of calculating the probability of fuzzy events. Two methods have been suggested for this: the first from Zadeh [26] and the second from Yager [25]. Zadeh's formulation is as follows:

$$P(A) = \int_{R^N} U_A(x)dp = E\,U_A(x)$$

U_A is the membership function of the fuzzy set A, and $U_A \in [0,1]$. Yager argues that "... it appears unnatural for the probability of a fuzzy subset to be a number." We would further argue that Zadeh's formulation does not truly provide a probability of a fuzzy event but something quite different: the expected truth value of a fuzzy event. Yager proposes that the probability of a fuzzy event be a fuzzy subset (fuzzy probability):

$$P'(A) = \bigcup_{\alpha=0}^{1} \alpha \left[\overline{P(A_\alpha)}\right]$$

where α specifies the α - level subset of A and since $P(A_\alpha) \in [0,1]$, P(A) is a fuzzy subset of [0,1]. This fuzzy subset then provides a probability of A for every α - level subset of A. Thus, depending on the required (or desired) degree of satisfaction, a probability of the fuzzy event A is available. In our case the desired level of truth is that which maximizes the bias of this event to the hypothesis. For example, if we wish to set a degree of satisfaction for the proposition "x is tall" and we are primarily interested in whether x is a basketball player then we wish to choose an α level which allows us to best differentiate BB players from non-BB players. We define this optimal α - level to be:

$$\max_{\alpha} \left|W(H:E_\alpha)\right| \quad \alpha \in [0,1]$$

$W(H:E_\alpha)$ is the weight of evidence of the strong α - level subset E_α provided towards the hypothesis H. The α -level which maximizes the bias of a fuzzy event to a

hypothesis (or null hypothesis) is the optimal α - level for minimizing systematic noise in the event.

The identification of important sets of symptoms or characteristics is done commonly by human medical experts and other professionals. For example, the combination of (abdominal pain, vomiting, fever) may indicate appendicitis with a certain probability or level of confidence. Our motivation for trying to discover important symptom groups is twofold: to identify which groups are important in a predictive sense and to quantify how important a group is. Also a factor in the decision of using symptom groups instead of individual variables is the belief that there exists many high order dependencies in this and other real-life domains. An symptom set may be of any size between 1 and N where N is number of possible symptoms. To find all such symptom sets requires an exhaustive search of high combinatorial complexity. This may be reduced somewhat by not examining groups that contain a subset of symptoms which are very rare. For example, if freq(High pain, low pulse) is very low then we need not look at any groups containing these two symptoms. Our present implementation examines sets up to size three. The weight of evidence of each symptom group is measured from the data and a test of significance decides whether this group has a weight significantly different from 0. Of significant interest is the clinician's endorsement of the important symptom sets that this method found. Those sets which showed as being important using the weight of evidence are symptom groups that the clinician would also deem as being significant.

3.4 Implementation and Results

Implementation was on a Sequent parallel processor with 4 Intel 80386's. The method was coded in C and Pascal and made much use of a programming interface to the ORACLE RDBMS. This provided a powerful blend of procedural and non-procedural languages in a parallel programming environment.

- Data was obtained for a training set of 253 equine colic cases each composed of 20 clinical variables. Also included for each case are several pertinent diagnostic codes: clinician's decision, presence of a surgical lesion, and lesion type. The prototype system provides a prediction for the presence of surgical lesions. Veterinary experts commonly have problems in differentiating between surgical and non-surgical lesions. Of primary concern to the clinicians is the negative predictive value that is, how often a surgical lesion is properly diagnosed. If a surgical lesion is present and is incorrectly diagnosed then the lesion is usually fatal for the horse. Presented below is a summary of our results using 89 cases from the training set.

From these results we can see that the method of evidence combination achieved an accuracy for negative prediction which exceeded the clinician's. Incorrect diagnoses are being reviewed by the clinician to see if some explanation can be found. There seems to be no correlation between clinician's errors and the computer technique's. This perhaps indicates that the clinician is adept at cases which are difficult for our techniques (and vice versa).

The following example shows our results for a case which had a surgical lesion but was not operated on by the clinicians. The horse displayed the following symptoms:

- 6 months old	High rect temp
Very high pulse	high resp rate
Cool temp at xtrem	Reduced per pulse
Norm mucous mem	Cap refill - 3s
Depressed	Hypomotile
Mod abdom distension	Sl nasogastric reflux
No reflux	Elevated Reflux PH
Normal rectal Xam	Distended lg intestine
Norm packed cell vol	Normal total protein
serosang centesis	High abd Tot Protein

and the method determined that the following evidence was *important*:

Evidence Towards Surgical Lesion:	
Symptom Group	**W(H:E)**
Adult,Hypomotile,Mod abdom dist	1.053
Sl nasogast,Dist L.I.,Norm Tot Prot	0.861
V high pulse,Red per pulse, C ref - 3s	0.861
Cool temp Xtrem,No reflux,Normal PCV	0.661
High rect temp, depressed	0.372
Serosang abdominocentesis	0.312
Evidence Against Lesion:	
H resp rate,norm mucous mem, norm rect Xam	-0.547

Final Results:

Prior Log Odds	0.530
- W(H:E)	3.573
Post Log Odds	4.103

- - - p(surg lesion) 0.804

Comparison of Predictive Power (89 Cases)		
Method	Negative Predictive Value	Positive Predictive Value
Clinician*	87.6%	100%
Weight of Evidence	96.7%	86.6%

* Figures obtained from these 89 cases. Previous studies have shown these values to be 73% and 93% respectively over a large sample.

For this case, this method strongly supports the surgical lesion hypothesis. It is interesting to compare these results to that of the classical regression model. Using this model the probability of a lesion, p, is predicted by: $p = \frac{e^Y}{(1-e^Y)}$, where

$$Y = 7.86 - 1.73(A2) - 1.54(ln(pulse)) - 0.498(Distension)$$

In this case A2 was 1 because the horse had a distended large intestine, the pulse was 114, and the distension was 3 (moderate). Substituting into the formula we get:

$$Y = 7.86 - 1.73 - 1.54(ln(114)) - 0.498(3)$$
and when we solve for p we find that p = 0.5818

Thus the classical regression analysis produces a p value greater than .5, but which is not strongly conclusive.

It is interesting to combine these findings with the results of Bayesian classification. This case belongs to autoclass 4 (determined with outcome information excluded). In this class, 80.1% of the 22 cases had a lesion, close to the value predicted but the weight of evidence formula. However, in this class only 19% of the cases lived and only 15% of the animals which had a lesion and were operated on actually lived. Pathology variables were particularly important for determining that *class and abdominal distension* was only moderately significant (although the doctors and logistic regression rely on this variable). The significant variables found by the PLS1 algorithm were also of very high weight for class 4. (One measurement of a pathology variable necessary for making a diagnosis was missing. The other factors indicated a slight preference for the presence of a lesion). The Autoclass information suggests a poor outcome prognosis in any case and indeed this was a situation in which the clinicians decided on euthanasia. Regression and weight of evidence techniques alone would not have suggested this decision.

4.0 Conclusions and Further Work

This paper has attempted to provide an idea of the methods being used to extract information from data in the development of an information system for medical diagnoses. Several techniques have been presented and some initial comparisons made.

Some tests of performance were accomplished by keeping aside portions of the test data. A new data set is currently being gathered with 168 cases which will be used to both test the methodologies and then refine the present diagnostic results. This new data set has been difficult to retrieve as not all the data used in the original set was on-line. We are taking care to ensure the information taken from hard copy records is entirely consistent with the first training set.

We are also looking at a more standard Bayesian model and trying to understand the dependencies and other conditioning in the data. The methodologies used here also help shed some light on this. The works by J. Pearl [6] and Spiegelhalter [10] are being considered for this approach.

In terms of developing an actual system using the methodologies of part 3, a prototype has already been implemented which considers symptom groups up to a size of three. A more advanced algorithm which tests independence between groups and provides an error estimate is currently being implemented in a blackboard architecture.

Terminals exist in the hospital in the work areas used by the clinicians and we are now proceeding to make the results available on incoming cases. Any final system would provide the doctors with selected information from several methodologies. This is to help especially with the diagnosis of difficult cases - as the real question is not just to be statistically accurate a certain percentage of the time, but to provide diagnostic aids for the harder cases.

Acknowledgements

The authors wish to thank the Ontario Veterinary College computing group, Drs. A. Meek and Tanya Stirtzinger (OVC) and Ken Howie for their help, especially with data collection. The Statistical consulting group at the University of Waterloo (C. Young) helped with some analyses. The encouragement of Dr. D.K. Chiu is also gratefully acknowledged.

References

1. Adlassnig, K.P., *Fuzzy Set Theory in Medical Diagnosis*, IEEE Transactions on Systems, Man and Cybernetics, vol. 16, 1986, pp. 260-265.
2. Berge, C., *Graphs and Hypergraphs*, North Holland Press, 1973.
3. Buchanan, B. and Shortcliffe, E., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, 1986.
4. Cheeseman, P.C., *Learning Expert Systems for Data*, Proc. Workshop of Knowledge-Based Systems, Denver, December 1984, pp. 115-122.
5. Cheeseman, P.C., Kelly, J., Self, M. and Stutz, J., *Automatic Bayesian Induction of Classes*, AMES Report, 1987.
6. Ducharme, N., Pascoe, P.J., Ducharme, G. and Lumsden, T., *A Computer-Derived Protocol to Aid in Deciding Medical or Surgical Treatment of Horses with Abdominal Pain*.
7. Ducharme, N., Ducharme G., Pascoe, P.J. and Horney, F.D., *Positive Predictive Value of Clinical Explanation in Selecting Medical or Surgical Treatment of Horses with Abdominal Pain*, Proc. Equine Colic Res., 1986, pp. 200-230.
8. Good, I.J., *Probability and the Weighting of Evidence*, New York: Hafner, 1950.
9. Good, I.J., *Weight of Evidence, Corroboration, Explanatory Power, Information, and the Utility of Experiments*, JRSS B,22, 1960, p. 319-331.
10. Lauritzen, S.L., Spiegelhalter D.J., *Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems*, accepted for JRSS, Series B.
11. Matthews, D. and Farewell, V., Using and Understanding Medical Statistics, Karger Press, 1985.
12. McLeish, M., Cecile M. and Lopez-Suarez, A., *Database Issues for a Veterinary Medical Expert System*, The Fourth International Workshop on Statistical and Scientific Database Management, 1988, June 88, pp. 33-48.

13. McLeish, M., *Exploring Knowledge Acquisition Tools for a Veterinary Medical Expert System*, The First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, June 1988,p778-788.

14. Cecile, M., McLeish, M. Pascoe P., Taylor, W., "Induction and Uncertainty Management Techniques Applied to Veterinary Medical Diagnosis", accepted for the 3rd AAAI Uncertainty Management Workshop, August, 1988.

15. Minsky, M. and Selfridge, O. G., *Learning in Random Nets*, Information Theory (ed. Colin Cherry; London Butterworths). p. 335-347.

16. Pearl, J., *Distributed Revision of Composite Beliefs*, Artificial Intelligence, Oct. 1987, p.173-215.

17. Popper, K.R., *The Logic of Scientific Discovery*, London: Hutchinson, 1959.

18. Quinlan, U.R., *Learning Efficient Classification Procedures and Their Applications to Chess End Games*, in Machine Learning: An Artificial Intelligence Approach , edited by R. Michalski, Tioga, 1983. pp. 463-482.

19. Rendell, L.A., *A New Basis for State-space Learning Systems and a Successful Implementation*, in Artificial Intelligence , vol. 4, 1983, pp. 369-392.

20. Rendell, L.A., *A General Framework for Induction and a Study of Selective Induction*, in Machine Learning, vol. 2, 1986, pp. 177-226.

21. Schwartz, W., Patil, R. and Szolovits, P., *Sounding Board, Artificial Intelligence in Medicine*, New England Journal of Medicine, vol. 16, no. 11, 1987, pp. 685-688.

22. Self, M. and Cheeseman, P., *Bayesian Prediction for Artificial Intelligence* , Proceedings of the Third AAAI Workshop on Uncertainty Management, 1987, pp. 61-69.

23. Spiegelhalter, D.J. and Knill-Jones, R.B., *Statistical and Knowledge-based Approaches to Clinical Decision Support Systems*, JRSS, B, 147, p35-77.

24. White, N.A., Moore, J.N., Cowgil, L.M. and Brown, N., *Epizootiology and Risk Factors in Equine Colic at University Hospitals*, Proc. Equine Colic Res., vol. 2, 1986.

25. Yager, R. R., *A Note on Probabilities of Fuzzy Events*, Information Sciences 18, 1979, p.113-129.

26. Zadeh, L. A., *Probability Measures of Fuzzy Events*, J. Math. Anal. Appl. 23, 1968,p. 421-427.

Methods of Approximate Reasoning in Expert Systems: Computational requirements

by

Ambrose Goicoechea

Department of Information Systems and Systems Engineering
School of Information Technology and Engineering
George Mason University

ABSTRACT

This paper presents a comparative study of six major, leading methods for reasoning: (1) Bayes' Rule, (2) Dempster-Shafer theory, (3) Fuzzy Set Theory, (4) MYCIN Model, (5) Cohen's System of inductive probabilities, and (6) a class of Non-monotonic reasoning methods. Each method is presented and discussed in terms of theoretical content, a detailed numerical example, and a list ostrengths and limitations. Purposely, the same numerical example is addressed by each method so that we are able to highlight the assumptions and computational requirements that are specific to each method in a consistent manner. Guidelines are offered to assist in the selection of the method that is most appropriate for a particular problem

KEY WORDS: Inference models, expert systems, imperfect knowledge, uncertainty, decision support systems, inference network, evidential reasoning.

1. INTRODUCTION

Intelligent systems that support human judgement and choice are computer-implemented procedures that seek to combine knowledge about a domain (e.g., problem or situation) with methods of conceptualizing, structuring and reasoning about such a domain. They incorporate, additionally, "formal" methods of reasoning about the domain that need to be brought to bear when tasks are poorly understood and structured, or when the information available is incomplete, fragmented, or otherwise imperfect. The ability of such computer-based expert systems to be able to look at part of the "picture" available and then make inferences about the true nature of the problem rests upon a knowledge base that is able to combine pieces of information available and utilize appropriate reasoning methods. Such reasoning methods include the heuristics or informal "rules of thumb" that people use to rapidly find solutions to problems, as well as formal reasoning methods that are useful in resolving problems about which experiential familiarity is slight.

A major motivation for this paper is the need to assess the progress in the development of methods that utilize imperfect information, and algorithms that offer the potential for utilization in computer-based expert systems. Many papers in the literature discuss one method, but few consider in a comparative manner two or more methods. As a result, the reader must deal with imperfect information about the capabilities, applicability and limitations of each method.

This paper borrows from other works in many respects. One of the numerical examples used throughout this paper, for instance, is originally due to Lee, Grize and Dehnad (1987), who have demonstrated four of the methods, specifically: (1) Bayes' Rule, (2) Dempster-Shafer, (3) Fuzzy Set theory, and (4) the MYCIN model. This paper presents their treatment of the first three methods, proceeds to expand on the description of and example for the MYCIN model, and two new examples are constructed to illustrate (5) Cohen's System of inductive probabilities and (6) a class of Non-monotonic reasoning methods. During the construction of the example for the non-monotonic reasoning method, valuable insight into the method was provided by the previous work of Cohen, Watson and Barret (1985) who present a realistic application to image analysis. Also, a comparative analysis by Black and Eddy (1985) has helped greatly in the discussion of the strengths and limitations of each method.

Interested readers are invited to write to the author for a complete version of this paper.

Consider the following diagnostic problem due to Lee et al. (1987):

"John seems to have a runny nose and irritated eyes. How likely is it that he is suffering from (1) a common cold, or from (2) a nostril allergy?"

2. BAYES' RULE

The problem would restated as follows:

IF: a person X has a runny nose, and X has irritated eyes,

THEN: conclude that X has only a common cold with probability p_1, and conclude that X has only a nostril allergy with probability p_2, AND conclude that X has neither a cold nor an allergy with probability p_3, AND conclude that X has both a cold and an allergy with probability p_4.

Also, we let the evidence be

E: X has a runny nose and irritated eyes, and let the set of hypotheses be

H_1: X has only a common cold

H_2: X has only a nostril allergy

H_3: X has neither a cold nor an allergy, and

H_4: X has both a cold and an allergy.

By Bayes' Rule we have that

$$p_1 = P(H_1|E) = P(E|H_1)\ P(H_1)|C$$

where

$$C = P(E|H_1)P(H_1) + P(E|H_2)P(H_2) + P(E|H_3)P(H_3) + P(E|H_4)P(H_4).$$

Strengths and Limitations. There are number of significant drawbacks in applying Bayes' rule to expert systems:

(1) the rule requires all the hypotheses to be disjoint and, in a large expert system, dividing a solution space into mutually exclusive subsets may be expensive;

(2) in the event of altering the probability of an event in a system (by adding or removing hypotheses) we would need to recalculate all the probabilities;

(3) there is no guarantee that the set of probabilities built into an expert system is consistent and coherent; for example, the product $P(A|B)P(B)$ may or may not be equal to $P(B|A)P(A)$;

(4) in realistic situations evidentiary information can quickly translate into very long sums and products of conditional and marginal distributions requiring substantial storage and computing resources.

3. THE DEMPSTER-SHAFER THEORY OF EVIDENCE

This theory of mathematical evidence (Shafer, 1976; Dempster, 1967) is basically a set-theoretic generalization of Bayesian theory.

There are some problems in the theory that are yet to be addressed in greater detail:

(1) Dempster's rule of combination cannot be applied in situations where there are considerable disagreements among the evidence, that is, when the cores of two belief functions are disjoint;

(2) in relistic cases a long chain of inferences may make the theory very inconvenient and expensive to use because of the increasing complexity in the structure of the core of the belief functions;

(3) the numerical stability of the theory needs to be analyzed further; in some cases, small variations in the basic probability assignments can produce a large variation in the results.

4. VAGUENESS IN FUZZY SET THEORY

In contrast to probability and evidence theory as models for representing uncertainty, a theory of possibility was proposed by Zadeh (1978) to represent vagueness inherent in some linguistic terms.

Our problem is decomposed into two rules:

Rule 1: IF a person X definitely has a runny nose,
AND X definitely has irritated eyes,
THEN X probably has a common cold;

Rule 2: IF a person X definitely has a runny nose,
AND X definitely has irritated eyes,
THEN X may or may not have a nostril allergy.

These two rules make use of the term set T( ):

T( ) = [definitely not, probably not, may or may not, probably, definitely].

There have been a number of applications of fuzzy logic to expert systems, including SPII (Martin and Pradee, 1986), and REVEAL (Jones and Morton, 1982). Some observations on possible drawbacks:

(1) the maximum and minimum rules for disjunction and conjunction may cancel valuable information when fuzzy individual assignments to various pieces of evidence include one assignment that is very close to zero;

(2) membership functions are context-sensitive; for example, a "small" building can be bigger than a "big" house; generic membership functions, if applied blindly, can lead to misleading results;

(3) computational and storage requirements can be large whenever individual membership functions are non-linear, non-trivial; discrete

approximations of non-linear membership functions place a significant demand on computer storage and computational requirements.

5. COHEN'S SYSTEM OF INDUCTIVE PROBABLITIES

Among the several researchers who have noticed anomalies and paradoxes in the application of conventional Bayesian probability to inference in certain situations is the Oxford logician L.J. Cohen. In Cohen's system, "inductive probabilites" are assigned to alternative hypotheses (Cohen, 1980).

Cohen's system is congenial to a process called "induction by elimination", as one proceeds to use evidence as a basis for elimination of some hypotheses, such that the hypothesis resisting being classified as "false" by evidence is then considered to be "correct", at least tentatively.

Figure 1 presents an illustration of an application to an inferential task involving four hypotheses about a particular situation. Evidence is gathered resulting on n evidentiary points. Each evidentiary point can be thought of as a "test" to apply to each hypothesis; some hypotheses pass (P) this test while others fail (F). At level four, that is after E_1, E_2, E_3, and E_4 have been considered, the assessed inductive probabilities (IP) are as follows:

Hypotheses	Inductive Probabilities at level i=4
H_1	$IP(H_1 \mid evidence) = 2/n$
H_2	$IP(H_2 \mid evidence) = 3/n$
H_3	$IP(H_3 \mid evidence) = 1/n$
H_4	$IP(H_4 \mid evidence) = 2/n$

and so it appears that hypothesis H_2 is the one that the pieces of evidence E_1, E_2, E_3, and E_4 support the most, tentatively. Read Schum (1987) for an in-depth study of Cohen's system as it contrasts with Bayesian theory.

6. MYCIN CERTAINTY FACTORS

The MYCIN experiment of Shortliffe and Buchanan (1975) was originally applied to a subdomain of medicine where little reliable data is available, and a rigorous application of Baye's rule would be difficult if not impossible.

MYCIN's theoretical framework includes terminology such as "measures of belief", denoted MB, "measures of disbelief", denoted MD, and "certainty factors", CF. Formally, these are defined as:

MB(H,E) = the measure of the belief in the hypothesis H, given evidence E,

MD(H,E) = the measure of the disbelief in the hypothesis H, given evidence E,and

$$CF(H,E) = MB(H,E) - MD(H,E).$$

Since MB(H,E) is a number between 0 and 1, and MD(H,E) is also a number between 0 and 1, the certainty factor CF(H,E) is a number between -1 and +1. A positive CF indicates that there is more reason to believe a hypothesis than to disbelieve it. A negative CF means that a hypothesis is more strongly rejected than confirmed. A CF of zero, is a "don't know" value which tells us that a hypothesis is independent of some evidence. Measures of belief are used in an inference network such as the one shown in Figure 2 to propagate evidence, leading to a hypothesis.

7. NON-MONOTONIC REASONING

Monotonic systems of thought are such that beginning with an initial set of premises, the number of statements or theorems that have to be shown true (e.g., to be proven as true) increases monotonically (increases continouusly) over time as new axioms or premises are added on. This is generally the case for many traditional, axiomatic formal systems of reasoning.

By contrast, in non-monotonic systems of thought, the number of practical structures of argument and belief may increase as well as decrease over time. This may be so because new data may compel an analyst to conclusions. Humans become skilled at merging conflicting data into existing arguments or beliefs so as to regain consistency while minimally disrupting the book-keeping activities within such a system.

A key concept in implementing nOn-monotonic systems is that of dependency-directed backtracking. As data and constraints are added to a non-monotonic system, they are treated as valid until a contradiction is found; when and if a contradiction is found, the system rearranges the set of beliefs that are "IN" (e.g., considered to be valid, true), and the set of beliefs that are "OUT" (e.g., considered to be not valid, not true). Traditional systems, in the face of contradiction, must backtrack past the data that was added immediately prior to the contradiction and then search for a path that is free of contradictions. As a result, many dead ends are encountered with exhaustive searches before a consistent total set of beliefs found (if available, at all). In a non-monotonic system, only those beliefs that actually contribute to a contradiction need to be examined.

During the knowledge-representation part of the problem use is made of data structures called support lists. A support list (SL) justification for a statement has the form:

Statement#	---	Statement	-	(SL (inlist)(outlist))

Such a justification is a valid reason for belief in the statement if every statement in its inlist is believed to be true, and every statement in its outlist is not believed to be true. Two types of justifications are used most frequently:
(1) A premise justification has an empty inlist and an empty outlist, i.e., (SL()()). Nothing else needs to be demonstrated to ensure acceptance of a statement with such a justification. Observed data and (unquestioned) general principles might be treated this way. For example,

N-1 Person X has a runny nose SL()())

is automatically regarded as IN.
(2) A monotonic justification has a non-empty inlist, but an empty outlist, as in

N-2 Person X has a nasal congestion (SL(Person X has a runny nose) (nasal membranes are normal))

8. A COMPARISON OF THEORIES

Figure 3 depicts the format and content of the conclusions reached by some of these methods. Conclusions are not and cannot be identical given the different calculi employed by these methods. Table 1 presents some general observations on the computational and structural requirements of each method,

Alternative Hypotheses:

Evidence:	H1 cold only	H2 allergy only	H3 no cold, no allergy	H4 cold and allergy
E1: Runny nose	P	P	P	P
E2: Irritated eyes	P	F	F	P
E3: Test tesults of nasal tissue culture	F	P	F	F
E4: Itching of nose and throat	P	F	F	P
E5: Medication B stops itching	F	P	F	P
E6: Medication A alleviates runny nose	P	F	F	F
E7: Swelling of nasal membranes	F	P	F	P
E8: Medication A causes drowsiness	P	P	P	P

legend:
P: Pass
F: Fail

Figure 1. Hypothesis testing in diagnostic problem.

which can serve as guidelines for matching a given problem to the most appropriate method

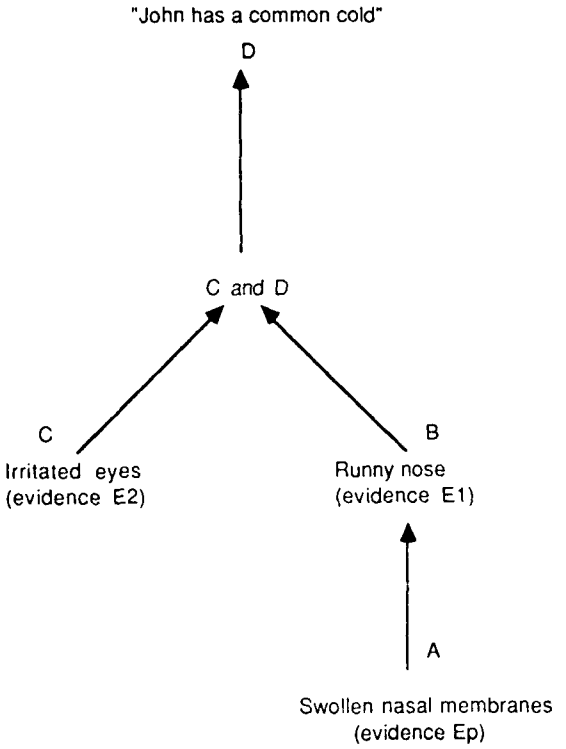


Figure 2. Inference network for diagnostic problem.

Acknowledgements-The author would like to thank Andy P. Sage who suggested this topic, guided portions of this study, and reviewed the manuscript reapeatedly, each time offering valuable comments. Many thanks also to David A.Schum who shared his knowledge and experience with Bayesian systems, alerted the author to the work of L.J. Cohen, and gave of his time, critique, and wonderful wit generously.

REFERENCES

[1] Lee, N.S., Y.L. Grize, and K. Dehnad, "Quantitative Models for Reasoning under Uncertainty in Knowledge-based Expert Systems", International journal of Intelligent Systems, Vol. 2, pp. 15-38, 1987.

[2] Cohen, M.S., S.R. Watson, and E. Barrett, Alternative Theories of Inference inin Expert Systems for Image Analysis, Technical Report 85-1, Decision Sciences Consortium, 7700 Leesburg Pick, Falls Church, Virginia, January 1985.

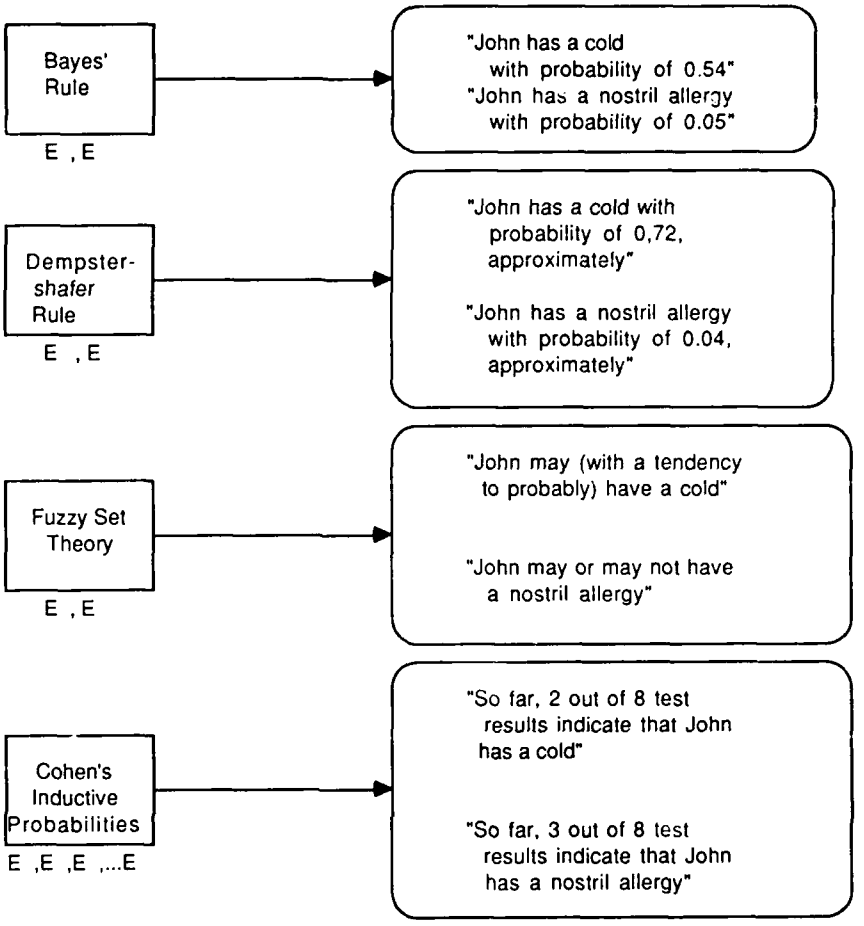
[3] Black. P.K., and W.F. Eddy, Models of Inexact Reasoning, Technical Report 351, Dept. of Statistics, Carnegie-Mellon University, Pittsburg, PA., May 1985.

[4] Shafer, G., A Mathematical Theory of Evidence, Princeton University Press, 1976.

Table 1. Comparison of Theories

Theories / Requirements	Baye's Rule	Dempster-Shafer	Fuzzy Set Theory	MYCIN Model	Non-Monotonic Reasoning	Cohen's Industive Probabilities
1. Theoretical Background	Strong	Strong	Strong	Light	Light	Moderate
2. Complexity of theory	Moderate	Strong	Strong	Low	Low	Low
3. Amount of information elicited	Moderate	Moderate	Large	Moderate	Moderate	Moderate
4. Number of Computations	Very Large	Moderate	Large	Moderate	Small	Small
5. Complexity of Computations	Low	Moderate	High	Low	Low	Low
6. Model Set Up	Moderate	Moderate	Moderate	Low	Moderate	Low
7. Computer Data Storage	Moderate	Moderate	High	Small	Small	Small
8. Ease of Application	Very Difficult	Difficult	Difficult	Easy	Moderate	Relatively Easy

Figure 3. Summary of Results



[5] Dempster, A.P., "Upper and Lower Probabilities Induced by a Mulit-Valued Mapping", Annals of Mathematical Statistics, Vol. 38, pp. 325-339, 1967.

[6] Schum, D.A., Evidence and Inference for the Intelligence Analyst, book in print, George Mason University, School of Information Technology and Engineering, Fairfax, Virginia, 1987.

[7] Sage, A.P., "Imperfect Information in Knowledge-based Decision Support Systems", Working Paper, George Mason University, Fairfax, Virginia, 1987.

[8] Zadeh, L.A., "Fuzzy Sets as a Basis for a Theory of Possibility", Fuzzy Sets and Systems, Vol. 1, No. 1, pp. 3-28, 1978.

[9] Cohen, L.J., "Bayesianism vs. Baconianism in the evaluation of Medical Diagnosis", British Journal of the Philosophy of Science, 1980.

[10] Martin-Clouaire, R., and H. Prade, SPII-1: A Simple Inference Engine of Accommodating both Imprecision and Uncertainty" in Computer-Assisted Decision Making, G.Mitra (ed.), North Holland, Amsterdam, The Netherlands, 1986.

[11] Jones, P., and P. Morton, "REVEAL", Computer-based Planning Systems, T, Naylor and M. Mann (eds.), Planning Executive Institute, Ohio, 1982.

[12] Shortliffe, E.H., and B.G. Buchanan, "A Model of Inexact Reasoning in Medicine", Mathematical Biosciences, 23, pp. 351-379, 1975.

Algorithms for Paired Comparison Belief Functions

David Tritchler, Ontario Cancer Institute and
the University of Toronto
Gina Lockwood, Ontario Cancer Institute

Abstract

This paper studies computational issues in applying the theory of belief functions to the method of paired comparisons. General algorithms are derived, and special cases depending on the focus of the analysis are studied.

KEYWORDS: Belief functions; Paired comparisons; Preference modelling.

1. Introduction

The paired comparison experiment is familiar in the social and decision sciences. A subject makes pairwise comparisons between members of a set $A = \{a_1, a_2, \ldots, a_n\}$ of n objects, choosing the most preferred object of each pair. There are several methods used to infer a preference relation, often a ranking, from the dominance data consisting of the stated choices, (see e.g. David (1963, 1971), Thompson and Remage (1964), and Flueck and Korsh (1975)).

Tritchler and Lockwood (1988) considered an extension of paired comparisons in which a certainty factor between 0 and 1 is expressed for each choice. They applied the theory of belief functions to study the weight given by this data to various preference relations, and derived various diagnostics describing the violation of transitivity and symmetry axioms by the subject.

The belief function analysis of paired comparisons is very time-consuming computationally. This paper studies the computational problem and derives algorithms. In section 2 we give a graph theoretic formulation of the belief function methodology. Section 3 poses the computing problem and gives an algorithm for computing basic probability numbers for the general case. In section 4 we give an algorithm for computing the beliefs for the 'best' object. Section 5 discusses Monte Carlo methods. Section 6 discusses computations for singleton focal elements.

2. Preliminaries

The essential framework of theory of belief functions was established by Dempster (1966, 1967 a,b, 1968 a,b, 1969). Shafer (1976) elaborated and extended the theory. The reader can consult Tritchler and Lockwood (1988) for a brief summary of the theory which is notationally consistent with the product space notation of Kong (1986) used in this paper, and for necessary concepts from graph theory. In this section we use a graphical representation to describe the application of the theory to paired comparisons. We begin with a frame of discernment $\Theta_{ij}=\{a_i \to a_j,\ a_j \to a_i\}$ for each pair of elements, where $a_i \to a_j$ indicates that a_i is preferred to a_j. A simple support function SUP_{ij} on Θ_{ij} is obtained from a comparison of a_i with a_j. If a_i was chosen with certainty r_i we obtain a basic probability function m whose focal elements are the subsets $\{a_i \to a_j\}$ and Θ_{ij}; $m(\{a_i \to a_j\}) = r_i$, and $m(\Theta_{ij}) = 1-r_i$.

If another comparison is made choosing a_j, and the resulting basic probability function $m(\{a_j \to a_i\}) = r_2$, $m(\Theta_{ij}) = 1-r_2$ is combined with the first, we obtain the orthogonal sum

$$m(\{a_i \to a_j\}) = r_1(1-r_2)/(1-r_1r_2),$$
$$m(\{a_j \to a_i\}) = r_2(1-r_1)/(1-r_1r_2),$$
$$m(\Theta) = (1-r_1)(1-r_2)/(1-r_1r_2).$$

The conflict between the two belief functions is $(1-r_1r_2)^{-1}$. In general, after combining any number of belief functions over the frame Θ_{ij}, the possible focal elements are $\{a_j \to a_i\}$, $\{a_i \to a_j\}$, and Θ_{ij} and we denote the orthogonal sum Bel^D_{ij}, with basic probability function m^D_{ij}. We denote the conflict among the simple support functions over Θ_{ij} by K^D_{ij}. Each of these focal elements has a canonical graph defined as follows: the canonical graph of a singleton focal element is that arc, and the canonical graph of Θ_{ij} is the graph with vertices $\{a_i, a_j\}$ and no arc. Thus, the subset ϕ_{ij} of Θ_{ij} corresponding to a canonical graph $G(\phi_{ij})$ is the set of all asymmetric graphs with node set $\{a_i, a_j\}$ which contain $G(\phi_{ij})$.

We define $\Theta(S)$ to be the product space $\Theta(S) = \Pi\Theta_{ij}$ where the product is over the indices $(i,j) \in S$ for $S = \{(i,j);\ 1 \le i < j \le n\}$; $\Theta(S)$ consists of all asymmetric relations (or graphs) with node set A. To combine the evidence from all of the pairs, each Bel^D_{ij} must be minimally extended from Θ_{ij} to $\Theta(S)$ giving $Bel^D_{ij}\uparrow\Theta(S)$ and then the orthogonal sum

$$Bel^D = Bel^D_{12}\uparrow\Theta(S) \oplus Bel^D_{13}\uparrow\Theta(S) \oplus \ldots \oplus Bel^D_{n-1\ n}\uparrow\Theta(S)$$

taken. The minimal extension of Bel^D_{ij} to $\Theta(S)$ assigns basic probability number m^D_{ij} to each set of the form $\phi_{ij} \times \Theta(S - \{(i,j)\})$, where ϕ_{ij} is a subset of Θ_{ij}. We define the canonical graph G of a focal element of the minimal extension $Bel^D_{ij}\uparrow\Theta(S)$ to have the same set of arcs as the canonical graph of the corresponding focal element of the marginal belief function Bel^D_{ij}; it will represent the set of all asymmetric graphs with node set A which contain G. We will write this set as

$(\Theta(S)|G)$. The focal elements of $Bel^{\rhd}$ are formed from intersections of focal elements of the $Bel^{\rhd}_{ij}\uparrow\Theta(S)$. Each focal element of $Bel^{\rhd}$ is of the form

$$\phi = \phi_{12} \times \phi_{13} \times \ldots \times \phi_{n-1\ n},$$

where ϕ_{ij} is a focal element of $Bel^{\rhd}_{ij}$ and each such set product defines a focal element of $Bel^{\rhd}$ with basic probability assignment

$$m^{\rhd}(\phi) = m^{\rhd}_{12}(\phi_{12})m^{\rhd}_{13}(\phi_{13})\ldots m^{\rhd}_{n-1\ n}(\phi_{n-1\ n})$$

(Tritchler and Lockwood, 1988). In graphical terms, the focal elements of $Bel^{\rhd}$ have canonical graph

$$G(\phi) = G_{12}(\phi_{12}) \cup G_{13}(\phi_{13}) \cup \ldots \cup G_{n-1\ n}(\phi_{n-1\ n}),$$

where $G_{ij}(\phi_{ij})$ is the canonical graph of ϕ_{ij}. Thus, intersections of focal elements of the $Bel^{\rhd}_{ij}$ are represented by unions of disjoint sets of arcs.

To introduce the assumptions of the linear ordering preference model we define a belief function $Bel^{\lhd}$ on $\Theta(S)$ with basic probability function $m^{\lhd}(L) = 1$ where L is the subset of $\Theta(S)$ consisting of all complete transitive irreflexive asymmetric graphs (linear orderings). Then $Bel = Bel^{\rhd} \oplus Bel^{\lhd}$ describes the subject's preference over A, constrained by our assumptions about the structure of his preferences.

Tritchler and Lockwood (1988) show that a focal element θ of Bel has canonical graph $G(\theta) = G^{t}(\phi)$, where ϕ is a focal element of $Bel^{\rhd}$ and $G^{t}(\phi)$ is the transitive closure of $G(\phi)$. Further, $m(\theta) = K^{\lhd} \sum m^{\rhd}(\phi)$ where the summation is over every focal element ϕ of $Bel^{\rhd}$ such that $G^{t}(\phi) = G(\theta)$, and $K^{\lhd}$ is the conflict between $Bel^{\rhd}$ and $Bel^{\lhd}$. $K^{\lhd}$ is an index of the subject's circularity since $K^{\lhd} = [1 - \sum m^{\rhd}(\phi)]^{-1}$, where the summation is over focal elements ϕ of $Bel^{\rhd}$ such that $G(\phi)$ contains a cycle.

We can represent a partial order in two ways; as a transitive, asymmetric graph H or alternatively, as a set $(L|H)$, the set of all linear extensions of H. The focal elements of Bel are of the form $(L|H)$, H a partial order. Also, $(L|G^{t}) = (\Theta(S)|G) \cap L$ is nonempty iff G is acyclic (Tritchler and Lockwood, 1988). This expression describes the intersection of a focal element of $Bel^{\rhd}$ with the focal element of $Bel^{\lhd}$.

For ease of exposition, we have formed $Bel^{\rhd}$ from the simple support functions corresponding to the comparisons in 2 steps: first we combined replications of the same comparison over Θ_{ij} and then the resulting $Bel^{\rhd}_{ij}$ were combined over $\Theta(S)$. However, Dempster's rule is commutative and the combination can actually take place in any order. The total conflict when computing Bel is $K = K^{\lhd} \cdot K^{\rhd}$, where $K^{\rhd} = \prod_{(i,j)\in S} K^{\rhd}_{ij}$ (Tritchler and Lockwood, 1988, Lemma 6).

3. Computing Basic Probability Numbers

We can dismiss the possibility of computing beliefs for all subsets of L, for both computational and interpretive reasons. The complexity of $2^{n!}$ (implied by the number of subsets) is clearly not feasible, and most of those subsets will have no interpretation as a relation. Tritchler and Lockwood (1988) show that each focal element can be interpreted as a partial order, and suggest that, for reasons of interpretability, we calculate beliefs and plausibilities only for subsets of L which correspond to a partial order. This indicates that we should calculate the basic probability numbers for the focal elements, and from them, calculate beliefs and plausibilities for selected partial orders. To this end, they characterize the focal elements which are contained in or intersect with a given partial order.

We can describe the calculation of the basic probability numbers in the following way. The analysis proceeds by first forming $Bel^{\rhd}$ and then combining $Bel^{\rhd}$ with $Bel^{\lhd}$. This can be done in two steps (Tritchler and Lockwood, 1988). At step 1, form the set Γ of all unions of the form $\cup_{(i,j)\in S} \Gamma_{ij}$, where Γ_{ij} is the canonical graph of a focal element of $Bel^{\rhd}_{ij}$. Γ consists of all focal elements of $Bel^{\rhd}$, represented by their canonical graph. At step 2, first the conflict $K^{\lhd}$ is calculated as the reciprocal of one minus the sum of the basic probability numbers for all elements of Γ containing a cycle; those elements are deleted from Γ; and the basic probability numbers of the remaining elements of Γ are normalized by the factor $K^{\lhd}$. Next the transitive closure of each element of Θ is computed and duplicates are eliminated, summing the basic probability numbers of identical closures. We denote the resulting set of graphs by Γ^{t}. This step corresponds to the orthogonal sum of $Bel^{\rhd}$ and $Bel^{\lhd}$. The complexity of the algorithm is $\prod_{(i,j)\in S} N_{ij}$, where N_{ij} is the number of focal elements of $Bel^{\rhd}_{ij}$.

The computation will be more efficient if we can induce the deletions of cycles and duplicates from Γ earlier. To this end, recall that each focal element ϕ of $Bel^{\rhd}$ can be written as $\phi = \phi_{12} \times \phi_{13} \times \ldots \times \phi_{n-1\ n}$ where ϕ_{ij} is a focal element of $Bel^{\rhd}_{ij}$. Suppose that $\phi_1 = \phi_{12} \times \phi_{13} \times \ldots \times \phi_{gh}$, for some g, h, $g<h<n$, is such that $G(\phi_1)$ contains a cycle. Then each focal element of $Bel^{\rhd}$ of the form $\phi = \phi_1 \prod_{(i,j)\in B} \phi_{ij}$ will be associated with a cycle for all choices of ϕ_{ij}, where ϕ_{ij} is a focal element of $Bel^{\rhd}_{ij}$, and $B = S - \{(1,2),(1,3),\ldots,(g,h)\}$. The collection of such focal elements will thus contribute total probability

$$M^{\rhd}(\phi_{12})\ M^{\rhd}(\phi_{13})\ \ldots\ M^{\rhd}(\phi_{gh})$$

to the probability mass for the null set when $Bel^{\triangleright}$ is combined with $Bel^{\triangleleft}$. Thus if we check for cycles while calculating $Bel^{\triangleright}$, we may prune focal elements with cycles as soon as they appear in the orthogonal sum.

Similarly, assume that
$\phi_1 = \phi_{12} \times \phi_{13} \times \ldots \times \phi_{1n}$ and
$\phi_2 = \phi_{12} \times \phi_{13} \times \ldots \phi_{1n}$ are are such that $G^{\triangleleft}(\phi_1) = G^{\triangleleft}(\phi_2)$. Then

$$\begin{aligned} G^{\triangleleft}(\phi_1 \prod_{(i,j)\in B} \phi_{ij}) &= (G(\phi_1) \cup G(\prod_{(i,j)\in B} \phi_{ij}))^{\triangleleft} \\ &= (G^{\triangleleft}(\phi_1) \cup G^{\triangleleft}(\prod_{(i,j)\in B} \phi_{ij}))^{\triangleleft} \\ &= (G^{\triangleleft}(\phi_2) \cup G^{\triangleleft}(\prod_{(i,j)\in B} \phi_{ij}))^{\triangleleft} \\ &= (G(\phi_2) \cup G(\prod_{(i,j)\in B} \phi_{ij}))^{\triangleleft} \\ &= G^{\triangleleft}(\phi_2 \prod_{(i,j)\in B} \phi_{ij}) \end{aligned}$$

for any choices of the focal elements of ϕ_{ij} of $Bel^{\triangleright}_{ij}$ for $(i,j)\in B$. Thus we may combine ϕ_1 and ϕ_2 into a single focal element with basic probability number $m^{\triangleright}(\phi_1) + m^{\triangleright}(\phi_2)$. In fact, the above arguments show that we may always represent e.g., ϕ_1 by $G^{\triangleleft}(\phi_1)$. This will identify duplicates and cycles (a cycle will result in $G^{\triangleleft}$ not asymmetric) at that step in the orthogonal sum.

4. Computing Beliefs About the Best Object

We can reduce the computational complexity of the analysis if we are interested in choosing the single best object, the best 2 objects, or in general the set of i most favoured objects. Let $Z_1, Z_2, \ldots, Z_k$ be the strong components partitioning of the dominance data. Define

$$I_m = \{(i,j);\ a_i, a_j \in Z_m,\ i < j\},\ m=1,\ldots,k$$

and let

$$I_0 = \{S - I_1 - I_2 - \ldots - I_k\}.$$

Define $Bel^{\triangleright}_m$ to be the orthogonal sum of the simple support functions over $\Theta(I_m)$, $m=1,\ldots,k$ and let

$$Bel_m = Bel^{\triangleright}_m \downarrow \Theta(S) \oplus Bel^{\triangleleft}.$$

Then

$$\begin{aligned} Bel &= Bel^{\triangleright} \oplus Bel^{\triangleleft} \\ &= Bel^{\triangleright}_0 \downarrow \Theta(S) \oplus Bel^{\triangleright}_1 \downarrow \Theta(S) \oplus \ldots \\ &\quad \oplus Bel^{\triangleright}_k \downarrow \Theta(S) \oplus Bel^{\triangleleft} \\ &= [Bel^{\triangleright}_0 \downarrow \Theta(S) \oplus Bel^{\triangleleft}] \oplus [Bel^{\triangleright}_0 \downarrow \Theta(S) \oplus Bel^{\triangleleft}] \\ &\quad \oplus \ldots \oplus [Bel^{\triangleright}_0 \downarrow \Theta(S) \oplus Bel^{\triangleleft}], \\ &= Bel_0 \oplus Bel_1 \oplus \ldots \oplus Bel_k \end{aligned}$$

by the idempotence of $Bel^{\triangleleft}$.

To restrict our attention to hypotheses about the most highly ranked objects, we define a partition of L:

$$T = \{T_1, T_2, \ldots, T_n\}$$

where
$T_i = \{\theta;\ \theta \in L$ and a_i is ranked highest in $\theta\}$.
T_j is a partial order with canonical graph $H_j = \{a_j \rightarrow a_i,\ i \neq j\}$.

The belief function of interest is $Bel\downarrow T$, the coarsening of Bel to the partition T. The computational complexity is reduced if we calculate $Bel_0\downarrow T \oplus \ldots \oplus Bel_k\downarrow T$, but we must verify that this calculation agrees with $Bel\downarrow T$. If it does, then we say that T discerns the interaction of the $Bel^{\triangleright}_i$, $i=0,1,\ldots,k$ relative to itself, using Shafer's terminology. Shafer and Logan (1987) give a criterion for assuring this discernment: for any choice of θ_i, $i=0,1,\ldots,k$ where θ_i is a focal element of Bel_i and any $T_j \in T$, $\theta_0 \cap \theta_1 \cap \ldots \cap \theta_k \cap T_j = \emptyset$ implies $\theta_i \cap T_j = \emptyset$ for some i.

Theorem 1. T discerns the interaction between Bel_0, Bel_1, ..., Bel_k relevant to itself.

Proof: We may write a focal element of Bel_i as

$$\theta_i = [\phi_i \times \Theta(S - I_i)] \cap L$$

for ϕ_i a focal element of $Bel^{\triangleright}_i$. Then by the independence of the frames $\Theta(I_i)$, $i=0,1,\ldots,k$,

$$\theta_0 \cap \theta_1 \cap \ldots \cap \theta_k = \phi_0 \times \phi_1 \times \ldots \times \phi_k \cap L$$

where $\phi = \phi_0 \times \phi_1 \times \ldots \times \phi_k$ is a focal element of $Bel^{\triangleright}$ with canonical graph $G(\phi) = G(\phi_0) \cup G(\phi_1) \cup \ldots \cup G(\phi_k)$ for $G(\phi_i)$ the canonical graph of ϕ_i.
First consider the case for which $\theta_0 \cap \theta_1 \cap \ldots \cap \theta_k = \phi \cap L = \emptyset$. Tritchler and Lockwood (1988, Theorem 1) show that $\phi \cap L = \emptyset$ iff $G(\phi)$ contains a cycle, which implies a cycle in some $G(\phi_i)$ by the definition of strong components, so $\theta_i = \emptyset$ and $\theta_i \cap T_j = \emptyset$.
Next consider the case $\theta = \theta_0 \cap \theta_1 \cap \ldots \cap \theta_k \neq \emptyset$, $\theta \cap T_j = \emptyset$. $T_j = (L|H_j) = (\Theta(S)|H_j) \cap L$. Let $\phi = (\Theta(S)|G)$ be any focal element of $Bel^{\triangleright}$ such that $\phi \cap L = \theta$. Note that $G = G(\phi_0) \cup G(\phi_1) \cup \ldots \cup G(\phi_k)$ for some choice of $\phi_0, \phi_1, \ldots, \phi_k$ where ϕ_i is a focal element of $Bel^{\triangleright}_i$. Then $\theta \cap T_j = \emptyset$ implies

$$\begin{aligned} \phi \cap L \cap T_j &= \phi \cap (\Theta(S)|H_j) \cap L \\ &= (\Theta(S)|G) \cap (\Theta(S)|H_j) \cap L = \emptyset \\ &= (\Theta(S)|G \cup H_j) \cap L = \emptyset, \end{aligned}$$

so $G \cup H_j$ contains a cycle. But then adding H_j to G creates a cycle, since G is acyclic. Thus G must contain some arc $a_k \rightarrow a_j$ incoming to a_j, where a_k is either in the strong component Z_m containing a_j or is in some other strong component. In the first case $G(\phi_m) \cap T_j = \emptyset$, and in the second case $G(\phi_0) \cap T_j = \emptyset$.

For a given strong component Z_m, the calculation of $Bel_m\downarrow T$ can be done over the frame $\Theta(I_m)$. To show this, explicitly express the operation of coarsening Bel_m to the partition T: $m_m\downarrow T(V) = \sum m(B)$ where the

summation is over all focal elements B of Bel_m such that $V=\{T_j;\ T_j \cap B \neq \emptyset\}$. We can write $B \cap T_j \neq \emptyset$ as

$$(L|G(B)) \cap (L|H_j) = (\Theta(S)|G(B)) \cap (\Theta(S)|H_j) \cap L$$
$$= (\Theta(S)|G(B) \cap H_j) \cap L \neq \emptyset,$$

which occurs iff adding H_j to G(B) creates a cycle. This condition is completely determined by arcs in $\Theta(I_m)$, and our calculations can be done over $\Theta(I_m)$.

There is a further result leading to the efficient calculation of Bel_o. It states that each Bel^p_{ij}, $(i,j) \in I_o$ can be coarsened to T before combining.

Corollary 1. $Bel_o \downarrow T = Bel^p_o \downarrow T$. Further, T discerns the interaction among $Bel^p_{ij} \uparrow \Theta(S)$, $(i,j) \in I_o$ relevant to itself.

Proof: $M_o \downarrow T(V) = \sum_{B \in W(V)} M_o(B)$,

where $W(V)=\{B;B$ a focal element of Bel_o such that $V=\{T_j;\ T_j \cap B \neq \emptyset\}\}$. Thus

$$M_o \downarrow T(V) = \sum_{B \in W(V)} K^L_o \sum_{\phi \cap L = B} M^p_o(\phi) = \sum_{\phi \in Z(V)} M^p_o(\phi),$$

where $Z(V)=\{\phi;\ \phi$ a focal element of Bel^p_o such that $V=\{T_j;\ T_j \cap \phi \cap L \neq \emptyset\}\}$, since $K^L_o = 1$ is the conflict over I_o. Noting that $T_j \cap L = T_j$, we see that $M_o \downarrow T(V) = M^p_o \downarrow T(V)$. The discernment of interaction follows from an argument similar to the second case of the proof of Theorem 1.

An algorithm based on the above results is:

ALGORITHM 1:
1° Calculate the commonality function Q_o for $Bel_o \downarrow T$
2° For each I_m, $m=1,2...,k$
3° Calculate the basic probability function m_m for Bel_m by the method of section 3.
4° Coarsen m_m to T (this can be done concurrently with step 3°).
5° Calculate Q_m for $Bel_m \downarrow T$ from $m_m \downarrow T$.
6° For each subset B of T, calculate

$$Q(B) \propto \prod_{i=0}^{k} Q_i(B).$$

7° Calculate $Bel \downarrow T$ and the associated plausibility function $Pl \downarrow T$ from Q.

The above algorithm calculates beliefs for all (2^n) subsets of T. If only certain subsets are of interest it might be more efficient to calculate the orthogonal sums using basic probability functions instead of commonality functions.

It is instructive to consider 1°, the calculation of Q_o, separately. Each Bel^p_{ij}, $(i,j) \in I_o$ must be a simple support function, otherwise both $a_i \to a_j$ and $a_j \to a_i$ are in the dominance data and a_i and a_j would be in the same strong component. For each $a_i \in A$, let us collect all the Bel^p_{ij} which prefer some a_j to a_i, coarsen them to T, and take the orthogonal sum (justified by Corollary 1). The result is a simple support function with focus $T - \{T_i\}$. We thus obtain simple support functions over T of the form $SUP_i(T - \{T_i\}) = S_i$, i=1,2,...n. By Corollary 1, their orthogonal sum will be $(Bel^p \oplus Bel^L) \downarrow T$.

The conflict between the SUP_i is one. To see this, note that a null intersection of focal elements of the SUP_i is possible only if $T - \{T_i\}$ is a focal element of Sup_i for i=1,2,...,n. But this implies that each a_i has an incoming arc in I_o, implying a cycle and thus contradicting the definition of strong components. Thus combining the SUP_i yields the commonality function with simple form

$$Q_o(B) = \prod_{S_i \in B} (1-S_i).$$

Further, by Barnett (1981),

$$Pl_o(B) = 1 - \prod_{S_i \in B} S_i.$$

Thus if the set of comparisons has no circularities, so $I_o = S$ and $Pl_o = Pl$, the computations for each subset of T are of complexity linear in n.

5. Monte Carlo Method

Let $H = \{H_1, H_2, \ldots, H_p\}$ be a set of partial orders which are hypotheses of interest. For example, we could have H = T, T defined as in section 4. We wish to approximate $Bel(H_i)$, $Pl(H_i)$, and the conflict K. Interpreting a focal element θ as a random subset with probability $m(\theta)$, the Monte Carlo procedure is apparent from the graphical formulation in section 3 and is given below as an algorithm.

ALGORITHM 2:
Initialize M=0 Z=0
1° Repeat N times: initialize R=∅, G=∅
2° For each $(i,j) \in S$:
3° With probability $m^p(\phi_{ij})$ randomly select a focal element ϕ_{ij} from the focal elements of Bel^p_{ij}.
4° Add $m^p(\theta_{ij})$ to the set R.
5° If ϕ_{ij} is a singleton add the corresponding arc to G.
6° Calculate G^t and $m = \prod_{r \in R} r$.
7° M = M + m.
8° If G^t contains a cycle add m to Z.
9° If G^t is cycle-free, then for $H_i \in H$, i=1,2,...p
10° If H_i is a subgraph of G^t, allocate m to $Bel(H_i)$.
11° If adding $G(H_i)$ to G^t does not create a cycle, allocate m to $PL(H_i)$.
12° Set $K = (M-Z)^{-1}$. K is the conflict.
13° For i=1,2,...p set $Bel(H_i) = K\ Bel(H_i)$, $PL(H_i) = K\ PL(H_i)$.

Step 10° is justified by Lemma 4 of Tritchler and Lockwood (1988). Step 11° is testing for a

non-null intersection of the focal element with the hypothesis H_i.

6. Computing Beliefs and Plausibilities for Rankings

Suppose our interest is focused on rankings over A, i.e. singleton subsets of L. We assume that we have approximated the conflict K by the method of section 5, or used Theorem 2 of Tritchler and Lockwood (1988) to simplify the exact calculation. We also assume that we have preprocessed the data so that for each frame Θ_{ij} we have at most two belief functions. One is a simple support function focused on $\{a_i \rightarrow a_j\}$, while the other is focused on $\{a_j \rightarrow a_i\}$. Let SUP_1, SUP_2,..., SUP_N be the simple support functions so defined, where $m_i(A) = S_i$ for A the focus of SUP_i over some frame Θ_{ki}. Then the commonality function for Bel is

$$Q = K\, Q^L \prod_{i=1}^{N} Q_i \ , \qquad (1)$$

where Q_i is the commonality function for $SUP_i \uparrow \Theta(S)$ and Q^L is the commonality function for Bel^L. For a singleton set $\theta \epsilon L$, $Q^L(\theta) = 1$, and

$$Q_i(\{\theta\}) = \begin{cases} 1-S_i & \text{if the focus of } SUP_i \text{ is the reversal of an arc in } G(\theta), \\ 1 & \text{otherwise.} \end{cases}$$

Thus, since $\{\theta\}$ is a singleton, $PL(\{\theta\}) = Q\{\theta\}$ is easily computed when $\theta \epsilon L$.

We can compute Bel for singletons more efficiently if we can identify those rankings with zero belief without actually calculating their belief. The following definition and theorem enable us to do this. A Hamiltonian path is a path of length n where n is the number of objects in A, and no object is on the path twice.

Theorem 2. ϕ is a focal element of Bel^P and θ a singleton focal element of Bel such that $G^t(\phi) = G(\theta)$ iff

1) $G(\phi)$ is acyclic and
2) there is a Hamiltonian path in $G(\phi)$.

Proof: First, suppose that acyclic $G(\phi)$ contains a Hamiltonian path. Thus for any pair (a_i, a_j), either a_i is reachable in $G(\phi)$ from a_j or vice versa. By Theorem 2 of Tritchler and Lockwood (1988), θ such that $G^t(\phi) = G(\theta)$, is a focal element of Bel, and by Theorem 5.4 of Harary et al (1965), $G^t(\phi)$, is the canonical graph of a linear ordering, i.e. a singleton focal element.
Next assume that θ is a singleton focal element of Bel and $G^t(\phi) = G(\theta)$. Clearly $G(\phi)$ is acyclic. To establish 2), we define the score s_i of a_i to be the number of arcs in $G(\theta)$ of the form $a_i \rightarrow a_j$. Since θ is a linear ordering

$$\{s_1, s_2, \ldots, s_n\} = \{0, 1, \ldots, n-1\}$$

$a_i \rightarrow a_j$ is in $G(\theta)$ (Moon, 1968, Theorem 9). Choose a_i and a_j so that $s_i = k$ and $s_j = k-1$, and assume $a_i \rightarrow a_j$ is not in $G(\phi)$. Since $a_i \rightarrow a_j$ is in $G^t(\phi)$ there must be a path $a_i \rightarrow a_k \rightarrow \ldots \rightarrow a_j$ in $G(\phi)$ by Theorem 5.4 of Harary et al (1965). But then $s_i - s_j > 2$ since $G^t(\phi)$ is transitive, a contradiction, so $a_i \rightarrow a_j$ must be in $G(\phi)$. Thus $G(\phi)$ contains a Hamiltonian path which corresponds to the ranking of the a_i in θ.

Since $\{a_i \rightarrow a_j\}$ can be a focal element of $Bel^P{}_{ij}$, and thus an arc in some $G(\phi)$, iff a_i was preferred to a_j on at least one occasion, we can enumerate Hamiltonian paths in the dominance data to find singleton focal elements of Bel.

When a Hamiltonian path W is found, Bel for the corresponding singleton focal element is calculated by enumerating all acyclic ϕ such that $G(\phi)$ contains W. Specifically,

$$Bel(\theta) = K \prod_{i \in P} S_i \prod_{j \in R} (1-S_j) = Pl(\theta) \prod_{i \in P} S_i$$

where $P = \{i;$ the focus F_i of SUP_i is an arc in $W\}$ and $R = \{i;$ the focus F_i of SUP_i is of the form $a_j \rightarrow a_i$, where a_i preceeds a_j on the path $W\}$. To see this, divide the simple support functions into 3 classes corresponding to P, R, and the complement C of $P \cup R$. By Theorem 2, for an intersection ϕ of focal elements of the SUP_i to satisfy $G^t(\phi) = G(\theta)$, the focal elements F_i, $i \epsilon P$ must be in the intersection. Also, no focal element from R can then be in the intersection since that would create a cycle in $G(\phi)$. Any combination of focal elements from C in the intersection gives $G^t(\phi) = G(\theta)$, since the transitive closure of the arcs in W determines a complete graph, so we have

$$Bel(\theta) = \prod_{i \in P} S_i \prod_{j \in R} (1-S_j) \sum_{\substack{F_1, F_2, \ldots, F_k \\ \cap F_i \neq \emptyset}} m(F_1) m(F_2) \ldots m(F_k),$$

where F_i is a focal element of SUP_i, $i \epsilon C$. Since the arcs corresponding to focal elements from C are in a subgraph of the cycle-free complete graph $G(\theta)$, no choice of $F_1, F_2, \ldots, F_k$ can yield a null intersection, so the above summation reduces to 1.

The problem of enumerating Hamiltonian paths is NP-hard. We can prune the search for Hamiltonian paths by using (1) to restrict our search of rankings of high plausibility. As each arc is added to a candidate path the partial product corresponding to (1) is checked. If it falls below a threshold α, the attempt to complete that path is abandoned. This restricts the search to rankings of plausibility greater than α.

Acknowledgements

The authors are grateful to Arthur Dempster

for helpful suggestions concerning this work. This research was supported by the National Cancer Institute of Canada.

References

Barnett, J.A., (1981), "Computational Methods for a Mathematical Theory of Evidence, Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, B.C., 868-895.

David, H.A. (1963), The Method of Paired Comparisons, London: Griffin.

David, H.A. (1971), "Ranking the Players In a Round Robin Tournament," Rev. Inst. Internat. Statist. 39, 137-147.

Dempster, A.P. (1966), "New Methods for Reasoning Towards Posterior Distributions Based on Sample Data", Ann. Math. Statist., 37, 355-374.

Dempster, A.P. (1967a) "Upper and Lower Probabilities Induced by a Multivalued Mapping", Ann. Math. Statist., 38, 325-339.

Dempster, A.P. (1967b), "Upper and Lower Probability Inferences Based on a Sample from a Finite Univariate Population", Biometrika, 54, 515-528.

Dempster, A.P. (1968a), "Upper and Lower Probabilities Generated by a Random Closed Interval", Ann. Math. Statist., 39, 957-966.

Dempster, A.P. (1968b), "A Generalization of Bayesian Inference (with discussion)", J.R. Statist. Soc. B., 30, 205-247.

Moon, J.W. (1968), Topics on Tournaments, New York: Holt, Rinehart and Winston.

Shafer, G. (1976), A Mathematical Theory of Evidence, Princeton: University Press.

Shafer, G. and Logan, R. (1987), "Implementing Dempster's Rule for Hierarchical Evidence", Artificial Intelligence, 33, 271-298.

Thompson, W.A. and Remage, R. (1964), "Rankings From Paired Comparisons", Ann. Math. Statist., 35, 739-747.

Tritchler, D. and Lockwood, G. (1988), "A Belief Function Approach to Paired Comparisons", unpublished manuscript.

Dempster, A.P. (1969), "Upper and Lower Probability Inferences for Families of Hypothesis With Monotone Density Ratios", Ann. Math. Statist., 40, 953-969.

Flueck, J.A. and Korsh, J.F. (1975), "A Generalized Approach to Maximum Likelihood Paired Comparison Ranking", Ann. Statist., 3, 846-861.

Harary, H., Norman, R.Z. and Cartwright, D. (1965), Structural Models: An Introduction to the Theory of Directed Graphs, New York: Wiley.

Kong, A.M. (1986), "Multivariate Belief Functions and Graphical Models", Ph.D. Thesis, Department of Statistics, Harvard University.

Fusion and Propagation in Graphical Belief Models

Russell Almond, Harvard University

ABSTRACT

This paper demonstrates the potential of graphical belief function models in decision problems. The working of a simple example problem illustrates the basic procedures involved in calculating marginal and conditional beliefs in a complex system. First, graphical modeling techniques break the example down into a series of small relationships, linked in a model hypergraph. Next the relationships between the attributes (variables) of the problem are expressed as belief functions. A simple procedure (Kong[1986b]) transforms the model hypergraph into a *tree of cliques*. This is a tree of "chunks" of the original problem; the information in each clique can be combined independently of all other cliques except its neighbors. Each node in the tree of cliques passes messages (expressed as belief functions) to its neighbors consisting of the local information fused with all the information that has propagated through the other branches of the tree. This propagation algorithm, along with the fusion algorithm given by the direct sum operator, can easily compute marginal beliefs, and can save considerable computational cost over the brute force approach. Finally, the paper explores new methodology for presenting the results of the computation.

Key Concepts: *Graphical Models, Belief Functions, Bayesian Models, Fusion and Propagation, Probability in Expert Systems, Triangulated Graphs.*

1. Attacking Large Problems

Many large problems, of a type that occur frequently in expert systems, involve a large number of variables and complex information about the relationships among those variables. These are not the classical statistical problems of estimating parameters from repeated observations, but instead require combinations of evidence from diverse sources to reach conclusions about the plausibility of certain events. Thus a decision maker, using one of these models, requires marginal information about certain events or groups of events. Graphical models are a clear and concise way of describing problems of many variables with dependencies among those variables. The variables, or attributes, become nodes in a model hypergraph and are joined by hyperedges. Only relationships among variables which all share a common hyperedge must be modeled, considerably simplifying both the modeling and the computational task. Graphical models have been studied by Pearl [1986a,1986b], Moussouris[1974], and Lauritzen and Spiegelhalter[1988] in the Bayesian case, and Kong[1986a], Shafer, Shenoy, and Mellouli [1986] and Shenoy and Shafer[1986] in the belief function case.

Probability distributions are not quite flexible enough to model the more complex interactions that can take place among attributes (variables) in one of these models. Belief functions, usually represented by the set function BEL, are a generalization of probability that allow ways to express total ignorance, Bayesian probability distributions, conditional probability distributions (likelihoods), logical relationships (production rules) and observations. All these diverse types of knowledge can be combined with a uniform fusion rule, the direct sum operator. Belief functions can be simply restricted to a smaller frame and easily extended to a larger frame without adding additional information. Shafer[1976,1982] develops the theory of belief functions and Kong[1986a] summarizes it. Belief functions provide a more flexible modeling tool than probabilities, but their computational cost rises quickly with the size of the problem. The problem must be subdivided into manageable chunks before it is solved. Thus graphical models and belief functions work well together.

For example, consider a problem with N attributes or variables, and M independent relationships among groups of those attributes. Let Θ be the discrete joint outcome space for those N variables, and the relationships among those variables be modeled as a belief function over Θ. To compute the combined belief function, $\mathrm{BEL}_{\mathcal{G}}$, which in turn yields marginal belief functions for any attribute or group of attributes of interest, the computational cost is $M \cdot 2^{|\Theta|}$. For the special case where all of the attributes are binary variables, the cost is $M \cdot 2^{2^N}$.

The high potential computation cost, due to the large size of Θ, makes the direct computation of $\mathrm{BEL}_{\mathcal{G}}$ impractical. However, the following strategy, which I have implemented, makes the computational tasks manageable:

1. Break up the problem using the *Graphical Modeling* and conditional independence assumptions, as described in Section 2.
2. Locally model relationships with *Belief Functions*. This process will be briefly described in Section 3.
3. Re-express the graphical model as a *Tree of Cliques* (see Dempster and Kong[1988] and Kong[1986b]). The tree of cliques will be described in Section 4.
4. *Propagate* and *Fuse* local information to find margins of the total belief function. This will be described in sections 5 and 6.
5. Now compute and examine any desired marginal belief function. This will be described in section 7.

These procedures reduce the computational costs dramatically. Returning to the above example, let m be the number of nodes in the Tree of Cliques ($m > M$ but only slightly), and k be the maximum number of neighbors of a clique in the tree. Furthermore, let C^* be the largest clique in the tree, n be the number of variables in C^*, and Θ_{C^*} be the outcome space associated with C^*. Then the computational costs are no more than $m \cdot k \cdot 2^{|\Theta_{C^*}|}$, or for the case of binary variables, $m \cdot k \cdot 2^{2^n}$. In most cases $n \ll N$; this reduces the size of the double exponential and yields a large savings in computational time. Furthermore, these computational costs are worst case figures, based on arbitrarily complex belief functions. In practice, with simple belief functions, the computational costs will be much smaller.

This paper will illustrate the strategy by following its application to a simple example. Consider the reasoning by which the Captain of a ship decides how many days late her ship will arrive in port. The first step in reasoning about the Captain's Decision is to define the attributes (variables) of the problem. The goal is to find the *Arrival delay*, or by how many days the ship will be delayed (for simplicity assume it will be an integer). This delay is the sum of two attributes: the *Departure delay* and the *Sailing delay*. Before the ship leaves port it could be delayed for *Loading* problems; a *Forecast* of foul weather could cause the Captain to delay departure; and *Maintenance* could cause the ship to sit at the dock. For simplicity assume that each of these three factors delay departure by one day. Therefore the total *Departure delay* could be up to three days. Similarly, bad *Weather* en route could cause delays, as could needing to make *Repairs* at sea. These delays contribute to the *Sailing* delays, again an integer number of days.

2. Graphical Models

Graphical models provide a way of organizing information about the relationships among variables in problem domains with many variables. Decision problems, diagnosis problems, fault trees, log-linear models, and expert systems all fall into this category. Thus the graphical model is a form of knowledge representation, and a graphical model design very much resembles relational data base design.

Breaking a large problem (the complete model) into a series of smaller problems is the essence of graphical modeling. A model hypergraph, $\mathcal{G}$, organizes the pieces of the large problem. Each node of the model hypergraph is an attribute or variable of the problem.

Each edge of the model hypergraph corresponds to a group of attributes that are related through some mechanism, modeled with the methods of Section 3. Any pair of attributes (nodes) which are not directly connected are assumed to be conditionally independent under the Markov conditions (see below). This last part is important. Any group of attributes that share a common hyperedge will have a belief function mechanism which models their relationship, yet any two nodes which are not directly connected, will have no such mechanism. Instead, their relation will be implicit in the way they are connected through other nodes. Thus the models of small problems with the graphical structure linking them from the model of the large problem. Furthermore, independence conditions may be easier to elicit from an expert than joint distributions(Pearl[1982]). Thus building a graphical model limits the size of the joint distributions (or complex mechanisms) which experts must provide.

For example, Figure 1 shows the model hypergraph for the Captain's Decision, with the first letter of each attribute name representing the node.

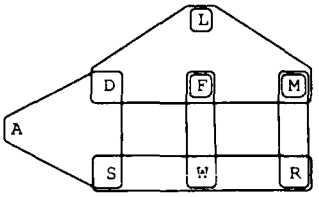


Figure 1. Model Hypergraph for the Captain's Decision

The second step in building a graphical model is to construct the edges. The edge containing, L, D, F and M represents the previously noted relationship between those attributes, namely that *Loading*, weather *Forecast* and *Maintenance* delays could each add one day to our total *Departure delay*. Similarly, the edge containing S, W and R expresses of a similar rule for delays that occur at sea. The edge containing A, D and S represents the logical assumption that the total delay equals the sum of the two partial delays. The edge containing F and W represents the condition that the *Weather* and the *Forecast* are dependent variables, and similarly the edge between M and R graphically depicts the dependence between *Maintenance* and *Repairs* at sea. Lastly the three singleton edges containing L, F and M respectively represent the Captain's prior beliefs about *Loading*, weather *Forecast*, and *Maintenance*.

There is a one to one correspondence between the edges of the hypergraph, and the component belief functions of the graphical model. Summing all of those belief functions, although computationally intensive, yields a complete picture of the interaction among all attributes of the problem. However, as a part of partitioning the total belief function into its components, we are also making certain assumptions about the conditional independence of the attributes. These are the *Markov conditions*, developed by Moussouris[1974] for the probabilistic case and extended here for belief functions and hypergraphs. (The formal definition of a belief function $\mathrm{BEL}(A)$ will follow in section 3. For the purposes of the discussion in this section, one can think of belief functions as some generalization of probabilities and still follow the arguments).

Markov Conditions. *Let $\mathcal{G}$ be a hypergraph. For any pair of nodes X and Y of $\mathcal{G}$, such that X and Y do not share a common hyperedge, let S be a set of nodes through which all paths from X to Y pass. For any such pair,* $\mathrm{BEL}(X \mid S,Y) = \mathrm{BEL}(X \mid S)$ *and* $\mathrm{BEL}(Y \mid S,X) = \mathrm{BEL}(Y \mid S)$. *That is X is independent of Y given S. Then $\mathcal{G}$ will be considered* a Markov Hypergraph *and is said to satisfy the* Markov Conditions.

The the belief functions of the Captain's Decision hypergraph (Figure 1) follow these Markov conditions. Consider the two attributes R and D. $\{S, F, M\}$ are one set of attributes which separate the two target attributes. Thus given S, F, and M fixed, R and D are independent. This makes sense in terms of the original model, since, if we know the *Sailing delay*, the *Forecast* for the weather, and the *Maintenance* record at the dock, it is plausible that the *Departure delay* and the *Repairs at sea* are independent.

Although the model hypergraph visually suggests many independence conditions, not all of the independence conditions fall neatly into the graphical model. For example, if the *Departure delay* is not fixed, it is plausible that the *Loading* and *Forecast* are independent. However, because their relationship represents one process, modeled with one belief function, they all share a common hyperedge. It is even possible to go further, and to model the relationship among several variables with a vacuous belief function (thus implying that they are totally independent). Although this makes the graphical model a less useful picture of the problem, it could be a useful technique for assessing the importance of dependency assumptions. Conversely, once two attributes are marked as conditionally independent in the graphical model, there is no way to add a dependence between them, without adding a new edge to the model.

3. Belief Functions

Here I will briefly discuss why belief functions are attractive tools for representing uncertainty in networks. Definitions and detailed descriptions can be found in Shafer[1976] or Kong[1986a]. I will only give an abstract of the ideas here.

Think of a set of possible outcomes $\Theta = \{\theta_1, \ldots, \theta_n\}$ of an experiment. Now given a subset A of the possible outcomes, define $\mathrm{BEL}(A)$ as the *belief* (a number between 0 and 1) that the true outcome will be in set A. With probabilities, one normally thinks of placing a mass function on the possible outcomes. With belief functions the *mass function*, $m(B)$, places mass on elements of the power set, 2^Θ, of the outcome space, that is on subsets of the possible outcomes. We normally restrict outselves to belief functions over discrete outcome spaces. The total mass is always one. For a normalized belief function, the mass on the empty set is always zero. Elements of the power set which have non-zero mass are called *focal elements*. Equation 1 relates the mass function to the belief function.

$$\mathrm{BEL}(A) = \sum_{B \subseteq A} m(B) \tag{1}$$

The *plausibility* of A, $\mathrm{PL}(A)$, is $1 - \mathrm{BEL}(\bar{A})$, where $\bar{A}$ is the complement of A with respect to Θ. Furthermore belief functions are superadditive; that is, if $A, B \subset \Theta$ and $A \cap B = \emptyset$, then $\mathrm{BEL}(A) + \mathrm{BEL}(B) \leq \mathrm{BEL}(A \cup B)$. Note that this last rule is a generalization of the corresponding case for probabilities.

The mass function of a belief function over a binary outcome space is particularly easy to interpret. For example, suppose the outcome space is $\Theta = \{\mathrm{F}, \mathrm{T}\}$ where F represents fair weather, and T represents foul weather. Consider the belief function with the following mass function:

$$\begin{aligned} m(\{\mathrm{F}\}) &= 0.6 \\ m(\{\mathrm{T}\}) &= 0.2 \\ m(\Theta) &= 0.2 \end{aligned} \tag{A.2}$$

We can interpret this as either: (1) There's a 20% chance of bad weather, a 60% chance of fair weather and 20% chance of unpredictable weather, or (2) There's a 20–40% chance of bad weather. In terms of betting odds, by this belief function I would be comfortable betting with odds better than 1:4 that there will be foul weather, or betting against foul weather with odds of better than 3:2. I am indifferent to (that is to say I would not take either side of) any bets within that region. It is useful to think of the mass placed on a given *focal element* (that is a subset of the outcome space which has non-zero mass) as the weight of evidence that suggests the outcome will be in the focal element, and that cannot be

divided (because of our ignorance) into finer divisions. As another example, we might think of the belief function given in Equation 2 as an urn containing black balls and white balls. With probability 0.2 we draw a black ball, with probability 0.6 we draw a white ball. With probability 0.2 we draw a ball which looks grey in this light, and we cannot determine its color without further experiments.

Belief functions have certain advantages over probabilities for modeling relationships in graphical models:

1. Upper and lower probabilities. A belief function provides a two-value assessment of uncertainty, the belief (BEL) or lower probability and the plausibility (PL) or upper probability, where a probabilistic model would provide a single number. Belief functions express uncertainty about the chance of an event occurring in simple way.

2. Can be used to represent: Bayesian probabilities, logical rules, observations, and ignorance. A Bayesian belief function has all its mass on the singleton subsets (representing the elements) of the outcome space. A "Vacuous" belief function, one with all its mass on the frame, Θ, provides an unambiguous definition of ignorance, unlike a so-called non-informative prior. A belief function with all of its mass on a single focal element can act as either an observation (if that focal element is a singleton) or a logical rule (if the focal element is more complex). Furthermore, dividing the mass between the frame and one other focal element produces a belief function which expresses partial support for a rule or an observation. (These operate in a way that the MYCIN (Buchanan and Shortlife[1984]) authors wanted their certainty factors to work). Belief functions incorporate both set and probability theory, and mix logical and probabilistic knowledge in a single uniform framework.

3. Belief functions easily marginalize *to smaller frames or* vacuously extend *to larger frames.* Merely projecting the focal elements unto a smaller or larger frame trivially marginalizes or extends a belief function. Note that the former is true of probabilities as well as belief functions, but the latter is not.

4. Fusion rule. The direct sum operator, $\oplus$, (Dempster[1968], Shafer[1976]) involves both set intersection and multiplication of probabilities with renormalization. It is a generalization of both logical and Bayesian inference rules.

Belief functions generally yield a great deal of flexibility at the cost of more complexity in both notation and computation. All the examples in (2) above are easy to specify. Furthermore for a binary variable, the belief function is easy to interpret through its upper and lower probability functions. The binary variable case seems to be, in general, a case were the Bayesian description yields too few parameters (we must choose a single probability) or else too many (we must specify a prior distribution over all possible probabilities that the value will be true).

On the other hand, as shown in section 7, a general belief function is often difficult to describe or interpret, simply because of the large numbers of sets of outcomes to examine. A belief function over a frame with n possible outcomes, has 2^n different outcome sets which could be assigned positive mass. Fortunately, for modeling purposes, graphical modelers can usually restrict themselves to the simple belief functions they do understand. If they can not, the problem may require further subdivision into simpler problems—another graphical modeling process.

Another aspect of the complexity of belief functions is that they are difficult to elicit from experts. While there is much literature on the general problem of eliciting probabilities from experts, no one as of yet has examined the problem for the more complex cases of belief functions. For the present, graphical modelers must rely on the simpler and more easily understood special cases.

If the input belief functions in the model are all Bayesian probabilities, then the marginal belief functions resulting from this computation will be probabilities too. Thus in a general way, every thing we discuss here applies to probabilities as well as belief functions. Doing the mathematics in the belief function notation helps us to understand what is happening without worrying about difficult technical details of extending probability distributions.

4. Making the *Tree of Cliques*

From the model hypergraph, $\mathcal{G}$, choose a collection of sets, $\mathcal{C} = \{C^1, \ldots C^m\}$, of $\mathcal{G}$'s attributes, $\mathcal{A} = \{A_1, \ldots A_n\}$. (Recall that attributes are nodes of the model hypergraph. I will deliberately use the term "attributes" here to avoid confusion with "cliques," the nodes of the tree of cliques.) If the model hypergraph is triangulated (acyclic), then the sets C^i will be the cliques (maximally complete subgraphs) of the model hypergraph. If the model is not triangulated, a procedure given in Kong[1986b] (also in Appendix II of Almond[1988]) produces these sets. The Kong procedure implicitly fills in hyperedges to create a triangulated graph; the C^i's are cliques of the triangulated graph. The Kong procedure also connects the cliques to form a tree, called the *Tree of Cliques*. The connections to satisfy a separation property which will be given below. For computational purposes, the tree of cliques is easier to use than the model hypergraph (Dempster and Kong[1988]).

A useful way to think about the tree of cliques is to consider each clique to be a group of attributes within which some complex interaction takes place. This complex interaction will be modeled by a belief function representing the information local to that clique, and by messages, also in the form of belief functions, passed from the neighboring cliques in the tree. Calculations are performed by propagating messages between the nodes via the schema given in Section 5 and by fusing the messages via the schema given in Section 6. The result is a belief function representing the margin of the graphical belief function for each of the margins C^i.

In order to make the computations more modular, we *augment* the tree of cliques by adding each of the original edges of the model hypergraph to $\mathcal{C}$ (new nodes in the tree of cliques) and connecting each new node to any clique that contains it. (Note that every hyperedge will always be contained in at least one clique). We can also augment the tree of cliques by adding (as a node) any set of attributes that is a subset of one of the cliques. In particular, the singleton sets of one attribute are always a subset of one of the nodes and are frequently margins of the graphical belief function that might be important to examine later on. The augmented collection of sets is called $\mathcal{C}^+$.

Each set, $C^i \in \mathcal{C}^+$, has a local belief function, BEL_{C^i}, representing the local information attached. This local belief function can be easily found from our graphical model, providing that that every edge of the original model hypergraph is in $\mathcal{C}^+$. For every node, C^i, in the augmented tree of cliques that corresponds to one of the original hyperedges, BEL_{C^i} is the belief function corresponding to that hyperedge. For every other node, BEL_{C^i} is vacuous.

As noted above, the tree of cliques is easy to build if the model hypergraph is triangulated. In order to get the tree of cliques from a non-triangulated graph, the Kong procedure fills in extra hyperedges to make a triangulated graph. There is often more than one way to fill in the hypergraph and different fill-ins lead to different trees of cliques, some of which are better than others. Because, as discussed in the first section, the cost of combining belief functions is exponential with the size of the largest clique, trees with smaller cliques will be better. The problem of finding the optimal tree of cliques is NP-hard. Kong and I have developed some heuristics for finding good trees of cliques that seem to do well. (Given in Appendix II of Almond[1988]).

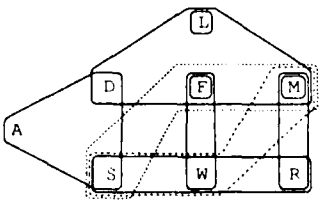


Figure 2. Model Hypergraph with fill-ins.

Let us return to the Captain's Decision problem. Figure 2 reproduces the model hypergraph from Figure 1. Notice that some new edges (the dashed lines) have been implicitly filled in as part of the construction procedure. Figure 3 shows the tree of cliques.

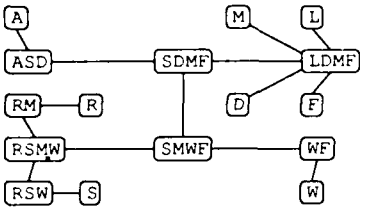


Figure 3. Augmented Tree of Cliques

The nodes $\{S, D, M, F\}$, $\{L, D, M, F\}$, $\{R, S, M, W\}$, $\{A, S, D\}$ and $\{S, M, W, F\}$ are the cliques of the graph in Figure 2. The nodes, $\{A, S, D\}$, $\{L, D, M, F\}$, $\{R, M\}$, $\{S, W, R\}$, and $\{W, F\}$ all correspond to original edges of the model hypergraph. These latter nodes are loaded with the belief functions corresponding to those edges. Furthermore, $\{M\}$, $\{L\}$, and $\{F\}$ all have univariate prior belief functions associated with them, which are also loaded into the appropriate cliques. The edges $\{S, D, M, F\}$, $\{S, M, W, F\}$ and $\{R, S, M, W\}$ are all associated with the filled in edges. They have vacuous belief functions associated with them. The remaining nodes, corresponding to the remaining singleton edges, also have vacuous belief functions associted.

The nodes, C^*, of the tree of cliques are connected so as to satisfy the separation property (Kong[1986b]). This is also true of the augmented tree of cliques.

Separation Property. *Given two nodes in* $\mathcal{C}^*$:

$$C^1 = \{A_{i_1}, \ldots, A_{i_n}\}$$
$$C^2 = \{A_{j_1}, \ldots, A_{j_m}\}$$

Let C^* *be any clique lying on the path between them. Then:*

$$C^1 \cap C^2 \subseteq C^*$$

The separation property also implies that the subgraph of the tree of cliques consisting of all cliques that contain a given attribute, or set of attributes it will be connected. This is not obvious, but can be seen after examining figure 3 for a few minutes.

5. Propagation

At the heart of the computational system, the cliques (nodes) of the tree pass messages among themselves (Dempster and Kong[1988]). This message passing system propagates the local information (which makes up the graphical model) to global information which can be used to answer questions about the process being modeled. Their system operates as follows.

The cliques pass "messages" to their neighbors in the tree of cliques, in the form of belief functions. Define $\mathrm{BEL}_{C^i \Rightarrow C^j}$ to be the belief function passed from C^i to C^j. Its frame corresponds to $C^i \cap C^j$. Each clique "fuses" its incoming messages with its local information, and "propagates" the results as its outgoing message. The fusion step will be described in detail in the next section.

When the node C^i has received messages from all its neighbors except C^j, it can calculate the message $\mathrm{BEL}C^i \Rightarrow C^j$ and pass it to C^j. Therefore the outermost leaves of the tree can immediately pass their messages inwards. The outermost cliques (the leaves of the tree) pass their information toward the center (root). When all the information reaches the center, the cliques in the center start passing messages back towards the outside (leaves).

Let us illustrate this with an example. Figure 4 shows the tree of cliques propagating their messages inwards, towards the root $\{S, M, W, F\}$. When $\{S, M, W, F\}$ receives all of its incoming messages, outward propagation can occur. The flow then follows backwards through the arrows of Figure 4.

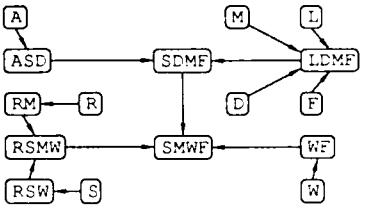


Figure 4. Propagating Inwards

Incidentally, there is nothing in this discussion which is specific to belief functions; belief functions only provide a convenient uniform notation for both the local information and the messages. This passing messages in a tree of cliques works equally well in the special case when all the belief functions are Bayesian probability distributions.

6. Fusion

Now let us turn to the details of what happens inside each clique to "fuse" incoming messages to produce the outgoing messages. At this moment, a single node, C^*, which has neighbors $C^1, \ldots, C^k$, has received messages from all but the last of its neighbors. This is shown in Figure 5.

The node C^* must now compute the message, $\mathrm{BEL}_{C^* \Rightarrow C^k}$, that is to be passed to the remaining clique. Equation (1) shows the calculation that C^* does.

$$\mathrm{BEL}_{C^* \Rightarrow C^k} = \left[\mathrm{BEL}_{C^*} \oplus \left(\bigoplus_{s=1}^{k-1} \mathrm{BEL}_{C^s \Rightarrow C^*} \right) \right] \qquad (1)$$

The message passed is just the sum of the incoming messages from all of the other cliques, $\mathrm{BEL}_{C^s \Rightarrow C^*}$'s, with the local information, BEL_{C^*}, stored at that clique.

Each of these calculations is done over the frame corresponding to C^*, and then the result is projected onto the frame corresponding to C^k. [Note: the computational cost is then $\leq m \cdot k \cdot 2^{|C^*|}$, as given in Section 1.] A clique can pass a message as soon as it has received messages from all but one of its neighbors. In particular, the leaves, which only have one neighbor, can immediately pass a message to that neighbor. Therefore the fusion and propagation algorithm is:

- Starting with the leaves, the nodes propagate messages inwards until the messages reach the root. At each successive stage, the inner nodes receive all of their incoming messages and can pass towards the center.
- At this point, the root has received messages from each of its neighbors, thus it can pass outwards in all directions, calculating its messages by equation (1); each time a clique calculates a message, it omits the destination from the sum. When a clique receives a message from the center, it now has all of its incoming messages and can send messages to each of its more outward neighbors. This is continued until the leaves receive their messages.

At this point each of the cliques has a series of incoming messages describing the contribution of the other parts of the tree to the total belief function. If we wish to view the margin of the total belief function, $\mathrm{BEL}_{\mathcal{G}}$, corresponding to a given clique, C^*, we simply sum all of the incoming messages with the local component, as shown in equation 2.

$$\mathrm{BEL}_{\mathcal{G}}|_{C^*} = \mathrm{BEL}_{C^*} \oplus \left(\bigoplus_{s=1}^{k} \mathrm{BEL}_{C^s \Rightarrow C^*} \right) \qquad (2)$$

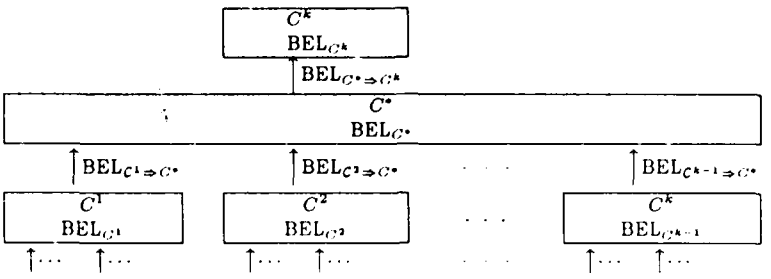


Figure 5. Messages passed to and from node C^*

The separation property, mentioned in Section 4, assures us that the marginalization done in each step does not lose any information. Thus it enables calculation over smaller frames, at considerable savings in time. That this procedure is correct follows from Kong[1986a,1986b].

A similar procedure can be used for sensitivity analysis. If we modify one of the component belief functions, the modified node in the tree of cliques simply passes new set of messages outwards to the others. The messages passed inwards toward the modified clique will be unchanged, and can be re-used. The same margins are examined before and after modification, and the change indicates the sensitivity of those marginal beliefs to the assumption represented in the changed component.

7. Examining the Results

The output of the fusion and propagation algorithm is the conflict, or the extent to which the belief functions are inconsistent, and the marginal belief functions for each clique (or singleton element) in the tree. For the Captain's Decision example, the belief function over *Arrival Delay* is of special interest. We will examine both the conflict and the *Arrival Delay* here.

The total *conflict* is mass that would be assigned to the null set, if the direct sum operator did not renormalize the belief functions. It varies from 0 to 1, and indicates to what extent the component belief functions are inconsistent, or how much mass is placed on contradictory cases. In this case the conflict is zero. Thus conclusions are formed from reinforcing positive evidence, rather than by ruling out contradictory possibilities.

Recall that *Arrival Delay* has seven possible values, $\Theta_A = \{0, 1, 2, 3, 4, 5, 6\}$. Belief functions are defined over the power set of the outcome space, so the belief function takes on 128 values. One way of cutting down the information to a manageable size is to only observe the *focal elements*; that is the sets of possible outcomes which are making a positive contribution to our belief (those elements with a non-zero mass). There are 21 of these as shown in table (1). The mass numbers represent support for the true outcome being in a given set (focal element). In our example, there is relatively small support for any given day, but there is large support for focal elements representing large sets of days.

The raw focal elements are difficult to interpret, as is generally true for non-binary variables. Let us try to make some meaningful summaries of the results.

First, look at the lower and upper expectations for the arrival time. Equation 3 gives the formula for calculating them.

$$E^*(A) = \sum_{B \subset \Theta_A} m(B) \min_{x \in B}(x) = 2.388$$

$$E_*(A) = \sum_{B \subset \Theta_A} m(B) \max_{x \in B}(x) = 0.824$$

That means that on average (in a series of hypothetical but never realized trials), the ship will be between 1 and 2 days late.

Mass	Focal Element	Mass	Focal Element
0.04	{0}	0.01	{4}
0.07	{1}	0.01	{4, 2}
0.16	{1, 0}	0.03	{4, 3}
0.04	{2}	0.03	{4, 3, 2}
0.01	{2, 0}	0.07	{4, 3, 2, 1}
0.12	{2, 1}	0.04	{4, 3, 2, 1, 0}
0.09	{2, 1, 0}	0.01	{5, 4, 3, 2}
0.02	{3}	0.01	{5, 4, 3, 2, 1}
0.02	{3, 1}	0.001	{5, 4, 3, 2, 1, 0}
0.06	{3, 2}	0.	{6, 5, 4, 3, 2, 1, 0}
0.04	{3, 2, 1}		$= \Theta_A$
0.10	{3, 2, 1, 0}		

Table 1. Focal elements on Arrival delay

Obviously, looking at beliefs and plausibilities for all 128 subsets of Θ_A would be exhausting, but looking at a carefully chosen batch of those subsets could reduce the task considerably. One such group of sets is the batch of singleton sets corresponding to each day. Figure 6 shows graphicly the beliefs and plausibilities for theses sets. The beliefs (lower probabilities) are the solid lines and the plausibilities (upper probabilities) are the dotted line.

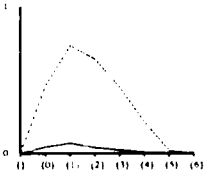


Figure 6. Single Day Beliefs for Arrival delay

Another group of interesting sets of arrival days is the batch of propositions that the ship will arrive before a certain day, or after a certain day. For example, {0, 1, 2} would represent less than 3 days and {3, 4, 5, 6} would represent at least 3 days. These are shown in Figures 7a and 7b. Because of the relationships between beliefs and plausibilities, Figure 7a is the same as Figure 7b when turned upside down and the dotted and solid lines are reversed.

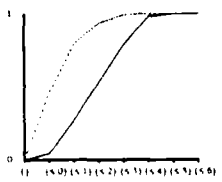


Figure 7a. Fewer than n days for Arrival delay

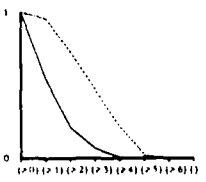


Figure 7b. Greater than n days for Arrival delay

Work on useful summaries is still at a very preliminary stage. The great flexibility that was an advantage in specifying the model can become complexity in examining the results. Clever ways of summarizing the resulting information then become necessary.

8. Conclusions and Future Directions

My experience with both the Captain's example, and the approximately 2,500 lines of LISP code that implement the general algorithms described here, suggests that Graphical Belief Models can successfully model uncertainty in practical decision problems. The algorithms mentioned here are easy to code and to use. Sensitivity analysis is simple in this system, and the theoretical framework is very attractive.

The simple example illustrates how to partition a problem into small pieces using graphical models, and how to model various relationships among the attributes. Re-arranging the model hypergraph into a tree of cliques allows calculation of the composite belief function's margins by a simple message passing algorithm. Finally, the simple belief function inputs give rise to the complex belief function shown in table 1. Although the calculation of $BEL(A)$ and $PL(A)$ for any given $A \subseteq \Theta$ is simple, this belief function may not be easy to comprehend and must be summarized. The methods used in section 7 are only a few of the ways we might interpret the results of the example

The system has a number of strengths:

1. The graphical modeling is simple to understand. It also forces the user to be precise about the relationships among the attributes. This may seem like a drawback to people who are used to adding rules in a fast and loose manner to a PROLOG database, but actually it forces one to model the interactions correctly in the planning stage, rather than tune them by extensive debugging. Of course sensitivity analysis methods can be applied for fine tuning or in cases where there is uncertainty about the model.
2. The belief functions are a flexible tool for modeling many types of relationships among the variables of a problem.
3. The fusion and propagation algorithm lowers the computational cost of calculation, making large problems tractable.

Unfortunately, the system also has some weaknesses. For large outcome spaces, a general belief function (such as the one given in table 1) can be a much more complex than a probability distribution over the same space. Furthermore, belief functions are unfamiliar objects and we do not have the wealth of methods for interpretation and anecdotal experience that we do with probabilities. The conflict that occurs in assembling the composite belief function is almost certainly a useful tool for discovering what is happening within a graphical model. On the other hand, we have very little experience with evaluating what conflict means. Work on the interpretation of belief functions is just beginning.

Although the fusion and propagation algorithm successfully breaks modest sized problems into tractable pieces, a very large problem, such as the fault tree analysis for a nuclear power plant, will require new approaches. Supercomputing might help. The direct sum operator is very amenable to vectorization, possibly taking advantage of hypercubic notations for sets, while the message passing algorithm described above would work well on a variable architecture machine, such as the connection machine. Furthermore, methods of modularizing graphs (breaking very large problems into more modestly sized pieces on which the fusion and propagation algorithm will work) to make large problems solvable on small machines need to be explored.

The next stage in this research is to use the system to work larger examples. This will help to develop both the theory and the algorithms to accommodate the new examples. The methods described here make belief function methodology accessible for practical problems; its real potential has yet to be realized.

Bibliography

Almond[1988]. "Fusion and Propagation in Graphical Belief Models: With Appendices." Technical Report S-121, Harvard University, Department of Statistics (A longer version of this paper).

Buchanan and Shortlife[1984]. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA.

Dempster[1968]. "A Generalization of Bayesian Inference." *Journal of the Royal Statistical Society, Series B,* Vol 30, pp 205-247.

Dempster and Kong[1988]. "Uncertain Evidence and Artificial Analysis." Technical Report S-120, Harvard University, Department of Statistics. (To appear in the *Journal of Statistical Planning and Inference.*

Kong[1986a]. "Multivariate Belief Functions and Graphical Models." Ph.D. thesis, Technical Report S-107, Harvard University, Department of Statistics.

Kong[1986b]. "Construction of a Tree of Cliques from a Triangulated Graph." Technical Report S-118, Harvard University, Department of Statistics.

Lauritzen and Spiegelhalter[1988]. "Fast Manipulation of Probabilities with Local Representations–With Applications to Expert Systems." *Journal of the Royal Statistical Society, Series B,*, Vol. 50.

Moussouris[1974]. "Gibbs and Markov Random Systems with Constraints." *Journal of Statistical Physics,* Vol. 10, pp. 11-33.

Pearl[1982]. "Fusion, Propagation, and Structuring in Belief Networks." *Artificial Intelligence,* Vol. 29, pp. 241-288.

Pearl[1986]. "Markov and Bayes Networks: A Comparison of Two Graphical Representations of Probabilistic Knowledge." Technical report R-46, University of California at Los Angles, Computer Science Department.

Shafer[1976]. *A Mathematical Theory of Evidence.* Princeton University Press.

Shafer[1982]. "Belief Functions and Parametric Models." *Journal of the Royal Statistical Society, Series B,* Vol. 44, pp. 322-352.

Shafer, Shenoy, and Mellouli[1986]. "Propagating Belief Functions in Qualitative Markov Trees." Working Paper No. 196, University of Kansas, School of Business.

Shenoy and Shafer[1986]. "Propagating Belief Functions with Local Computations." *IEEE Expert,* Vol. 1 No. 3, pp. 43-52.

Variants of Tierney-Kadane

G. Weiss & H. A. Howlader
University of Winnipeg, Winnipeg, Manitoba

Abstract

Bayes estimation of the reliability function of the logistic distribution under a log-odds squared error loss with a non-informative prior is considered by using the approximation method of Tierney & Kadane (1986). Direct application of the procedure does not yield correct results and so some variations of the procedure are considered.

1. Introduction

In Bayesian estimation it is often necessary to evaluate the ratio of two integrals which cannot be expanded into closed-form expressions. Numerical approximation of the ratio of the integrals is necessary. Two recent procedures to achieve this approximation have been proposed by Lindley (1980) and by Tierney & Kadane (1986). The Lindley (1980) procedure is fairly well-behaved and can be applied in all situations. The procedure of Tierney & Kadane (1986) works well in certain situations, such as, for computing the posterior means of the parameters of probability distribution, and gives results which are more accurate than the method of Lindley (1980). For approximations from small samples or over restricted spaces, however, the Tierney & Kadane (1986) procedure can be very erratic and may lead to incorrect results. See Howlader & Weiss (1987, 1988).

In particular, the method does poorly in cases where the integral in the numerator ranges over positive and negative values. In this paper, we consider one such instance, in the estimation of the reliability function of a logistic distribution.

2. Bayes estimation

The logistic probability density function may be written as

$$f(x|\mu, \sigma) = \frac{c}{\sigma} e^{-c\left(\frac{x-\mu}{\sigma}\right)} \Big/ \left[1 + e^{-c\left(\frac{x-\mu}{\sigma}\right)}\right]^2, \quad (1)$$

$-\infty < x < \infty$, $-\infty < \mu < \infty$, $\sigma > 0$, and where $c = \pi/\sqrt{3}$, μ is the mean, and σ is the standard deviation of the distribution.

Here, we consider the Bayesian estimation of the reliability function,

$$R_t = \frac{1}{1 + e^{-c\left(\frac{t-\mu}{\sigma}\right)}}, \quad (2)$$

by using the method of Tierney & Kadane (1986), using the non-informative prior $p(\mu, \sigma) \propto \frac{1}{\sigma}$. Combining this prior density with the likelihood function,

$$L(\mu, \sigma|\boldsymbol{x}) = \left(\frac{c}{2\sigma}\right)^n \prod_{i=1}^{n} (1 + \cosh \zeta_i)^{-1}, \quad (3)$$

the joint posterior density of μ and σ is

$$\pi(\mu, \sigma|\boldsymbol{x}) \propto \sigma^{-(n+1)} \prod_{i=1}^{n} (1 + \cosh \zeta_i)^{-1}, \quad (4)$$

where $\zeta_i = c\left(\frac{x_i - \theta}{\sigma}\right)$.

For most statisticians, interested mainly in controlling the amount of variability, it has become standard practice to consider squared-error loss functions. In the case of estimating a reliability, the usual squared-error loss does not seem appropriate as the reliability, which is a probability, is contained in the closed interval [0, 1], and hence the 'distance' from the true value is bounded. One remedy, is to first compute the log-odds ratio of the probability, which maps the [0, 1] interval onto the entire real line. It would thus be reasonable to use the squared-error of the log-odds,

$$\text{Loss}(r_t, R_t) = \left[\log\left(\frac{r_t}{1 - r_t}\right) - \log\left(\frac{R_t}{1 - R_t}\right)\right]^2. \quad (5)$$

The Bayes estimator, R_t^* of R_t under this loss function is the value of r_t which minimizes the posterior risk, $E[\text{Loss}(r_t, R_t)|\boldsymbol{x}]$, such that

$$\log\left(\frac{r_t}{1 - r_t}\right) = E\left[\log\left(\frac{R_t}{1 - R_t}\right)\middle|\boldsymbol{x}\right] = \delta, \quad (6)$$

which gives

$$R_t^* = \frac{1}{1 + e^{-\delta}}. \quad (7)$$

Then,

$$\delta = E\left[\log\left(\frac{R_t}{1 - R_t}\right)\middle|\boldsymbol{x}\right] = E\left[c\left(\frac{\mu - t}{\sigma}\right)\middle|\boldsymbol{x}\right]$$

$$= \frac{\int\int c\left(\frac{\mu - t}{\sigma}\right)\sigma^{-(n+1)} \prod_{i=1}^{n} (1 + \cosh \zeta_i)^{-1} d\mu\, d\sigma}{\int\int \sigma^{-(n+1)} \prod_{i=1}^{n} (1 + \cosh \zeta_i)^{-1} d\mu\, d\sigma}. \quad (8)$$

Tierney & Kadane (1986) gave a method of evaluation of the ratio of integrals, such as the posterior mean of a function $u(\theta)$, which has the form

$$\mathbf{E}[u(\theta)|x] = \frac{\int u(\theta)\pi(\theta|x)\,d\theta}{\int \pi(\theta|x)\,d\theta}. \tag{9}$$

by writing

$$\ell = \frac{1}{n}\log \pi(\theta|x) = \frac{\log p(\theta) + \log \mathbf{L}(\theta|x)}{n},$$

and (10)

$$\ell^* = \ell + \frac{1}{n}\log u(\theta) = \frac{\log u(\theta) + \log p(\theta) + \log \mathbf{L}(\theta|x)}{n}.$$

Thus, (9) takes the form

$$\mathbf{E}[u(\theta)|x] = \frac{\int e^{n\ell^*}\,d\theta}{\int e^{n\ell}\,d\theta}. \tag{11}$$

Tierney & Kadane (1986) claim that their method derives from an approximation method due to Laplace. Whereas Lindley (1980), in a similar approximation, expands both the numerator and the denominator of (11) about a common point [the mle or posterior mode], Tierney & Kadane (1986) expand each integral seperately about the point which maximizes the integrand. This method requires only the first and the second derivatives of the posterior density. Following Tierney & Kadane (1986), the equation (11) in the multi-parameter case takes the approximate value

$$\hat{\mathbf{E}}[u(\theta)|x] = \sqrt{\frac{|\Sigma^*|}{|\Sigma|}}\,exp\left\{n[\ell^*(\hat{\theta}^*) - \ell(\hat{\theta})]\right\}, \tag{12}$$

where $\hat{\theta}^*$ and $\hat{\theta}$ maximize ℓ^* and ℓ, respectively, and Σ^* and Σ are negatives of the inverse Hessians of ℓ^* and ℓ at $\hat{\theta}^*$ and $\hat{\theta}$, respectively.

To apply the method in the logistic case, we need to maximize

$$\ell = -(1 + \tfrac{1}{n})\log \sigma - \tfrac{1}{n}\sum \log(1 + \cosh \varsigma_i),$$

and, (13)

$$\ell^* = -\tfrac{1}{n}\log[1+e^{\eta}] - (1+\tfrac{1}{n})\log\sigma - \tfrac{1}{n}\sum\log(1+\cosh\varsigma_i),$$

where $\eta = c\left(\frac{t-\mu}{\sigma}\right)$.

Setting $\frac{\partial \ell}{\partial \mu}$ and $\frac{\partial \ell}{\partial \sigma}$, respectively to zero produces the system

$$\sum \varphi(\varsigma_i) = 0, \text{ and } \sum \varsigma_i \varphi(\varsigma_i) = n+1, \tag{14}$$

which produces the posterior mode. Also,

$$\frac{\partial^2 \ell}{\partial \mu^2} = \frac{-c^2}{n\sigma^2}\sum \gamma(\varsigma_i),$$

$$\frac{\partial^2 \ell}{\partial \mu \partial \sigma} = \frac{-c}{n\sigma^2}\sum\left\{\varphi(\varsigma_i) + \varsigma_i\gamma(\varsigma_i)\right\},$$

and, (15)

$$\frac{\partial^2 \ell}{\partial \sigma^2} = \frac{1}{n\sigma^2}\left\{(n+1) - \sum\left\{2\varsigma_i\varphi(\varsigma_i) + \varsigma_i^2\gamma(\varsigma_i)\right\}\right\}.$$

Similarly, setting $\frac{\partial \ell^*}{\partial \mu}$ and $\frac{\partial \ell^*}{\partial \sigma}$, respectively to 0 gives

$$\sum \varphi(\varsigma_i) = -\xi, \text{ and } \sum \varsigma_i \varphi(\varsigma_i) = n+1-\eta\xi, \tag{16}$$

where $\xi = \left(1 + e^{-\eta}\right)^{-1}$

Here,

$$\frac{\partial^2 \ell^*}{\partial \mu^2} = \frac{-c^2}{n\sigma^2}\left\{\tfrac{1}{2}\gamma(\eta) - \sum \gamma(\varsigma_i)\right\},$$

$$\frac{\partial^2 \ell^*}{\partial \mu \partial \sigma} = \frac{-c}{n\sigma^2}\left\{\xi + \tfrac{1}{2}\eta\gamma(\eta) + \sum\left\{\varphi(\varsigma_i) + \varsigma_i\gamma(\varsigma_i)\right\}\right\}, \tag{17}$$

$$\frac{\partial^2 \ell^*}{\partial \sigma^2} = \frac{1}{n\sigma^2}\left\{(n+1) - \sum\left\{2\varsigma_i\varphi(\varsigma_i) + \varsigma_i^2\gamma(\varsigma_i)\right\} - \left\{2\eta\xi + \tfrac{1}{2}\eta^2\gamma(\eta)\right\}\right\}.$$

The procedure of Tierney & Kadane (1986) is difficult to apply directly. The procedure requires that the integrals in (11) be strictly positive, and the procedure should only be applied when the integrals are not near zero. In (8), δ will be positive if R_t is greater than $\frac{1}{2}$ (*i.e.* $t > \mu$), otherwise it is negative.

If the value of t is fixed at some value away from the mean, μ, then the integrand will be either positive ($t > \mu$) or negative ($t < \mu$), almost surely. In this case, the procedure can be applied by applying the method to the positive integrand (taking absolute value) and determining the sign afterwards. If the value of t is fixed at some point near the mean, however, this procedure will not work.

The estimators of the reliability function for 10,000 simulations of samples from (1) with $\mu = 25$ and $\sigma = 5$ were computed and the histograms constructed. Figure 1, the histogram for $t = 15$, shows the typical distribution of an estimator about the true mean, $R_{15} = 0.9741$. However, Figure 2, the histogram for $t = \mu = 25$, shows a bi-modal distribution with the estimates being pulled away from the true mean, $R_{25} = \frac{1}{2}$.

Although, most apparant when $t = \mu$, there are similar disturbances to the distributions of the estimators of R_t for other values of t also. In the following sections several variations of the method are given.

3. Variant 1

One way to remedy the above situation is to shift the value of the integrand by adding the integrand to a large positive constant, $\gamma \gg 0$, which is then removed from the result (or subtracting, if $t < \mu$):

$$\delta = \mathbf{E}\left[log\left(\frac{R_\cdot}{1-R_t}\right)\middle|x\right] = \begin{cases} \mathbf{E}\left[\gamma + log\left(\frac{R_t}{1-R_t}\right)\middle|x\right] - \gamma, & t \geq \mu, \\ \gamma - \mathbf{E}\left[\gamma - log\left(\frac{R_\cdot}{1-R_t}\right)\middle|x\right], & t < \mu. \end{cases} \quad (18)$$

This procedure will not alter the maxima of the integrals and should be invariant with respect to the size of the constant (as long as it is large enough). However, since we altering the value of the integrand we may have slightly different computational precision for different values of γ.

Tables I and II give the means and the mean squared-errors (MSE) of the sampling distributions of the reliability estimates for different values of γ, the shift parameter, for 1000 samples generated from a logistic distribution as in (1) with $\mu = 25$ and $\sigma = 5$.

As expected, slightly differing histograms were obtained for different values of the shift parameter, however, such differences were minimal for any reasonably large shift value (greater than 50). These differences were virtually eliminated when the convergence criterion for the iterative procedure was strengthened from $\epsilon = 10^{-3}$ to $\epsilon = 10^{-16}$.

4. Variant 2

In this particular instance it is possible to rewrite the expectation as a linear function of expectations:

$$\delta = \mathbf{E}\left[c\left(\frac{\mu - t}{\sigma}\right)\middle|x\right] = c\left[\mathbf{E}\left(\frac{\mu}{\sigma}\middle|x\right) - t\,\mathbf{E}\left(\frac{1}{\sigma}\middle|x\right)\right] \quad (19)$$

Now, the Tierney-Kadane procedure can be applied to each of these expectations, since for μ not near zero, each of the integrands will be away from zero. Now, however, we are performing seperate approximations, and hence this variation of the procedure is not numerically equivalent.

Although these variants of the procedure are not numerically equivalent, they did produce very similar histograms in the simulation studies. See Figures 2—4. Compare Figure 4, which shows the histogram for variant 1 with a shift parameter of 500, and Figure 3, which shows variant 2, with the histogram in Figure 2 obtained by direct application of the method (identical to variant 1 but with shift of 0).

Table I

Means of the distribution of T-K estimator for diffent shifts

γ	$t = 15$	$t = 20$	$t = 25$	$t = 30$	$t = 35$
0	0.98791	0.91606	0.49198	0.08057	0.01190
50	0.98377	0.89322	0.49216	0.10304	0.01595
100	0.98362	0.89280	0.49217	0.10345	0.01610
500	0.98349	0.89237	0.49218	0.10387	0.01623
5000	0.98343	0.89214	0.49218	0.10410	0.01629
100000, $\epsilon = 10^{-9}$	0.97060	0.85790	0.49274	0.13753	0.02885
100000, $\epsilon = 10^{-16}$	0.98345	0.89288	0.49218	0.10396	0.01626
R_t	0.9741	0.8598	0.5000	0.1402	0.0259

Table II

MSE's of the distribution of T-K estimator for diffent shifts

γ	$t = 15$	$t = 20$	$t = 25$	$t = 30$	$t = 35$
0	.0004710	.0078547	.0536627	.0077600	.0004717
50	.0005580	.0079320	.0288334	.0080704	.0005525
100	.0005595	.0079287	.0286412	.0080680	.0005539
500	.0005615	.0079177	.0284799	.0080577	.0005558
5000	.0005625	.0079181	.0283601	.0080589	.0005569
100000, $\epsilon = 10^{-9}$	.0009700	.0077780	.0196678	.0078457	.0009500
100000, $\epsilon = 10^{-16}$	.0005618	.0079143	.0284491	.0080543	.0005561
SE(MSE)*	.000018	.000122	.000356	.000103	.000015

(*) These are the standard errors of the MSE, which is approximately constant for all values of the shift parameter, except for a shift of 0 and for the shift of 100000 with $\epsilon = 10^{-9}$ for which the values are slightly different.

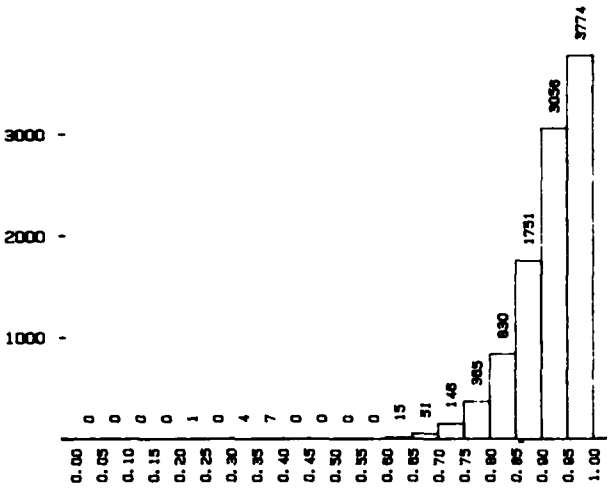


Figure 1: standard T-K under LOL at t = 20

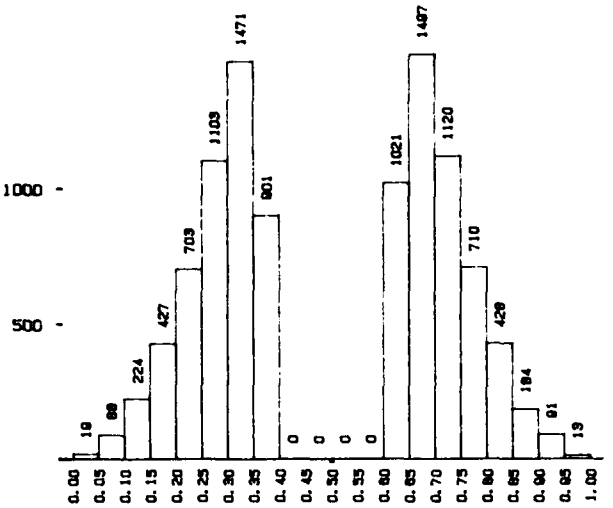


Figure 2: standard T-K under LOL at t = 25

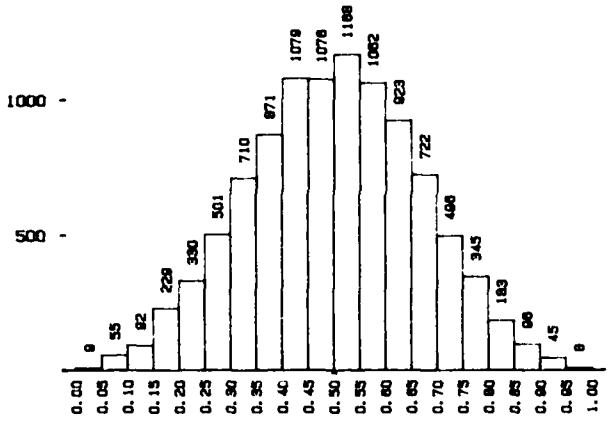


Figure 3: T-K variant 1 under LOL at t = 25

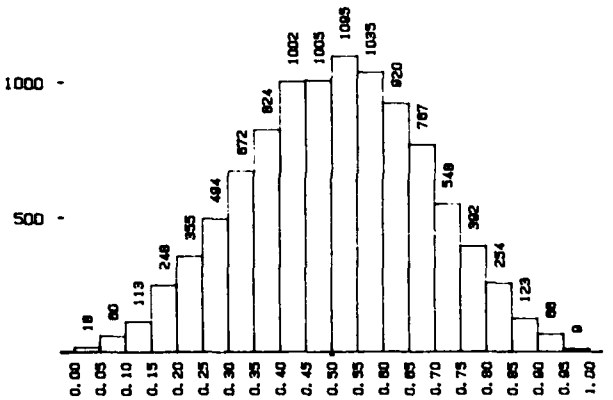


Figure 4: T-K variant 2 under LOL when t = 25

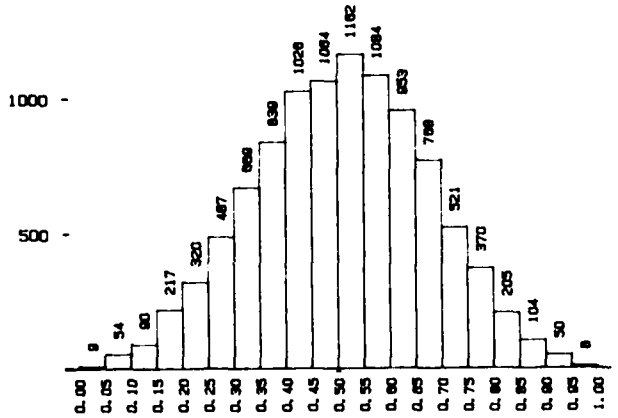


Figure 5: T-K variant 3 under LOL at t = 25

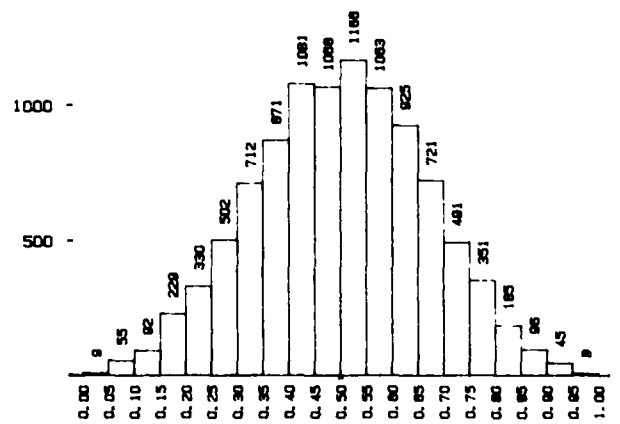


Figure 6: standard T-K under SEL at t = 25

The first variant, which is the technically more correct method, does require a seperate optimization of the numerator for each value of t (as in the standard T-K procedure). This second variant, however, is easier to apply and requires the optimization of only two numerators which can then be used to estimate the reliability function for all t. [Both variants also require the determination of the posterior mode, which is used to evaluate the normalizing integral in the denominator, which is, of course, common to all of the expectations].

What we would like is a method that has the advantages of both variants.

5. 'Variant' 3

Another variation of the procedure of Tierney & Kadane (1986) is suggested by the result in (19). In (19), we are estimating the reliability function (2) by replacing $1/\sigma$ and μ/σ in the kernel of reliability function by their respective posterior means. Again, since these cannot be obtained directly, each is approximated using the method of Tierney & Kadane (1986).

Although, this variation of the Tierney & Kadane (1986) procedure is only valid in this particular situation, it does suggest a method that might be generally applicable whenever we wish to approximate the Bayes estimator of a function of the parameters of the distribution.

The suggestion is to simply use the posterior means of the parameters themselves (μ and σ) and use these values in place of the unknown parameters. Again it may be necessary to compute these posterior means using the standard procedure of Tierney & Kadane (1986), or by direct computation when possible. This procedure is not really a variant of the Tierney & Kadane (1986), nor is it even a true Bayes procedure in that the estimator is not the minimum of the posterior expection (risk) of some loss function.

Recall that, for squared-error loss, the minimum posterior risk estimator (*i.e.*, Bayes estimator) is

$$\hat{R}_t^* = E(R_t|\boldsymbol{x}) = E\left[\frac{1}{1 + e^{c(t-\mu)/\sigma}}\middle|\boldsymbol{x}\right], \qquad (20)$$

while the minimum posterior risk estimator for log-odds squared-error loss is

$$\hat{R}_t^* = \frac{1}{1 + e^{c\left[tE[1/\sigma|\boldsymbol{x}] - E[\mu/\sigma|\boldsymbol{x}]\right]}}. \qquad (21)$$

We propose the estimator

$$\hat{R}_t = \frac{1}{1 + e^{c(t - E[\mu|\boldsymbol{x}])/E[\sigma|\boldsymbol{x}]}}, \qquad (22)$$

which is very likely will not be the minimum of the posterior expectation of any loss function, and is thus not a true Bayes estimator.

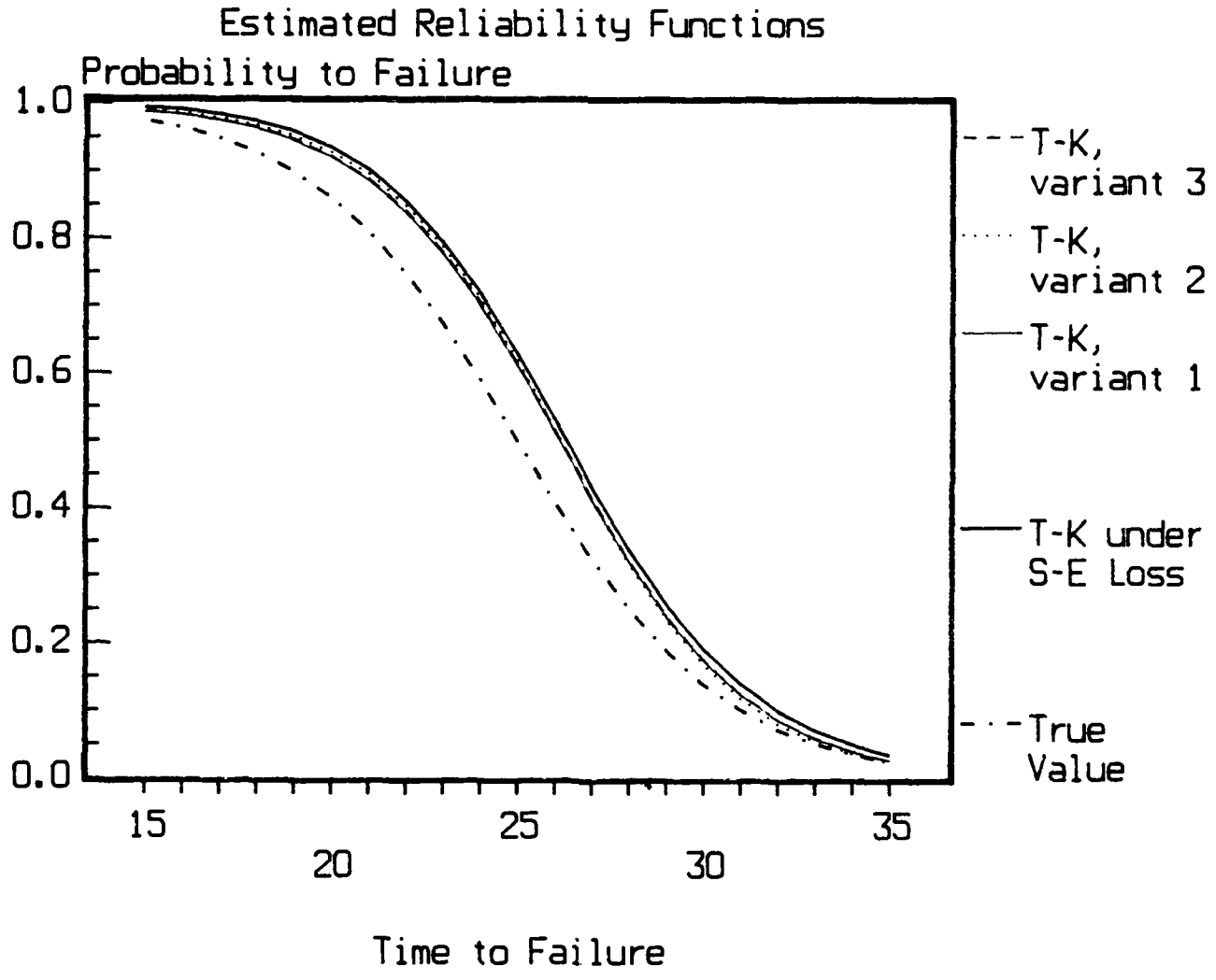
We generated the 10,000 samples of size 10 and constructed the histogram, shown in Figure 5, for the estimator in (22), for which the posterior expectations were approximated using the method of Tierney & Kadane (1986). Compare this distribution with the histograms for variants 1 and 2 shown in Figures 3 and 4 and also with the histograms for the Tierney & Kadane (1986) estimator under squared-error loss in Figure 6. There is not a very great difference betwen any of these. Hence, although (22) may not be a true Bayes estimator, as an *approximation* to a Bayes estimator it seems to be a justifiable alternative.

Also, the form (22) is intuitively attractive in that it suggests that, as in the case of the mle, for estimating a function $g(\theta)$, we can use the Bayes estimators of θ in place of θ in the functional form. As well, the method is generally applicable and requires a minimum number of optimizations.

Another comparison of these three variants is obtained by generating the estimated reliability curves for a single sample. A sample of 20 observations from the logistic distribution in (1) with $\mu = 25$ and $\sigma = 5$ is obtained as

10.372	20.570	21.204	21.540	24.118
24.256	24.325	24.357	25.301	25.344
26.661	26.881	26.989	27.377	29.110
29.450	29.888	31.849	31.946	37.473

Figure 7 shows the estimated reliability curves for the three variants under the log-odds loss, together with the curve for the true reliability and for the Tierney-Kadane estimator under squared-error loss. There again appears to be little difference between the four estimates. In particular, the curve for variant 3 is almost identical to that of variant 1.



Acknowledgment

This research is sponsored by grants from the Natural Sciences and Engineering Research Council of Canada.

References

Howlader, H.A., & Weiss, G. (1987) "Considerations on the approximation of Bayesian ratios of Integrals," ***Proceedings of the Nineteenth Symposium on the Interface***, Temple University, Philadelphia, PA.

Howlader, H.A., & Weiss, G. (1988) "Bayes estimators of the reliability of the logistic distribution," Techical Report #5, Department of Statistics, University of Winnipeg, Winnipeg, Manitoba.

Jeffreys, H. (1983) ***Theory of Probability***, Third edition. Clarendon Press, Oxford Science Publishers, Oxford, GB.

Kennedy, W., & Gentle, J. (1980) ***Statistical Computing***. Marcel Dekker, New York.

Lindley, D.V. (1980) "Approximate Bayesian methods. (with discussants)." ***Trabajos de Estadistica y de Investigacion Operativa***, 31: 232--245.

Tierney, L., & Kadane, J. (1986) "Accurate approximations for posterior moments and marginals," ***Journal of the American Statistical Association***, 81: 82--86.

X. NUMERICAL METHODS

NUMERICAL APPROACH TO NON-GAUSSIAN SMOOTHING AND ITS APPLICATIONS

Genshiro Kitagawa, The Institute of Statistical Mathematics

Abstract

A smoothing methodology for the analysis of time series is shown. The method is based on the general state space model which is expressed by conditional distributions. Various types of non-Gaussian models, nonlinear models and discrete variate models can be handled with this generic model. Recursive formulas for the prediction, filtering and smoothing of state are given for this general state space model. Unlike the familiar linear Gaussian state space model for which these formulas can be realized by simple Kalman filter and the fixed interval smoother, these formulas are implemented by using numerical expressions for related distributions. It thus becomes a computationally intensive method but is very flexible and is a useful tool for the analysis of time series that have been difficult to handle by the standard time series models. Many numerical examples are shown to illustrate the usefulness of the general state sapce modeling and of general smoothing algorithm.

1. Introduction

In time series modeling, we used to consider parameteric models. But with the spread of the application of the time series models, the limitation of the usual parametric models has been recognized and in some situations like seasonal adjustment, models with very many parameters were required. But obviously in such a situation, the ordinally maximum likelihood method does not work since it sometimes involves the estiamtion of more parameters than the number of observations.

For such a situation, penalized likelihood method might be applied. In this approach, the crucial problem is the selection of the tradeoff parameter λ. But there is a Bayesian interpretation of the problem based on smoothness prior, and we can determine the tradeoff parameter by maximizing the likelihood of the Bayesian model (Akaike, 1980). This smoothness prior method gave a solution to the many parameter problems, but this usually involves solution of linear equation with very large dimension. The use of the state space model mitigates this computational burden and makes the large parameteric model practical.

In the modeling of nonstationary time series, the main issue is the representation of time varying system. Linear Gaussian state space model are very useful for the modeling of gradual calnges of parameters. For example, seasoanl adjustment problem and estimation of changing spectrum can be treated with this linear Gaussian models. But these linear Gaussian models are not so adequate for the sudden changes or jump of parameterts. And we need another prior that allow sudden changes as well as gradual changes.

Such a prior can be well realized by the use of non-Gaussain state space model. But this non-Gaussain state sapce model can be further extended to a general state space model which can handle very wide situation including nonlinear and discrete distributions. In this paper, we derive recursive formulas of the prediction, filtering and smoothing for this general state space model. Unlike the familiar linear Gaussian state space model for which these formulas can be realized by simple Kalman filter and the fixed interval smoother, these formulas are implemented by using numerical expressions for related distributions. It thus becomes a computationally intensive method but is very flexible and is a useful tool for the analysis of time series that have been difficult to handle by the standard time series models. Many numerical examples are shown to illustrate the usefulness of the general state sapce modeling and of general smoothing algorithm.

2. General State Space Model and State Estimation

Consider a system described by a general state space model

$$\begin{aligned} x_n &\sim q(\cdot|x_{n-1}) \\ y_n &\sim r(\cdot|x_n), \end{aligned} \tag{1}$$

where y_n is the observation and x_n is the unknown state vector. q and r are conditional distributions of x_n given x_{n-1} and of y_n given x_n, respectively. The initial state vector x_0 is distributed according to the distribution $p(x_0|Y_0)$. The set of observations and the states, Y_m and X_m, are defined by $Y_m \equiv \{y_1, \ldots, y_m\}$ and $X_m \equiv \{x_1, \ldots, x_m\}$. The conditional distribution of x_n given X_k and the Y_m is denoted by $p(x_n|X_k, Y_m)$. The problem of state estimation can be formulated as the evaluation of $p(x_n|Y_m)$, the conditional distribution of x_n given observations Y_m. For $n > m, n = m$ and $n < m$, this formulates the problems of prediction, filtering and smoothing, respectively.

The above general state space model (1) implicitly assumes the following Markov properties:

$$\begin{aligned} p(x_n|X_{n-1}, Y_{n-1}) &= p(x_n|x_{n-1}) \\ p(y_n|X_n, Y_{n-1}) &= p(y_n|x_n). \end{aligned} \tag{2}$$

Obviously, our general state space model includes the

ordinary linear state space model

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n + w_n, \end{aligned} \tag{3}$$

with Gaussian white noises v_n and w_n.

Under the assumption of (2), it can be shown that the conditional distribution satisfies

$$p(x_n|x_{n+1}, Y_N) = p(x_n|x_{n+1}, Y_n). \tag{4}$$

In Kitagawa (1986,1987), it was shown that for the general state space model with (2) and (4), the recursive formulas for obtaining one step ahead prediction, filtering and smoothing distributions are (in continuous variate case) given as follows:

One step ahead prediction:

$$\begin{aligned} p(x_n|Y_{n-1}) &= \int_{-\infty}^{\infty} p(x_n, x_{n-1}|Y_{n-1})dx_{n-1} \\ &= \int_{-\infty}^{\infty} p(x_n|x_{n-1}, Y_{n-1})p(x_{n-1}|Y_{n-1})dx_{n-1} \\ &= \int_{-\infty}^{\infty} q(x_n|x_{n-1})p(x_{n-1}|Y_{n-1})dx_{n-1} \end{aligned} \tag{5}$$

Filtering:

$$\begin{aligned} p(x_n|Y_n) &= p(x_n|y_n, Y_{n-1}) \\ &= \frac{p(y_n|x_n, Y_{n-1})p(x_n|Y_{n-1})}{p(y_n|Y_{n-1})} \\ &= \frac{r(y_n|x_n)p(x_n|Y_{n-1})}{p(y_n|Y_{n-1})} \end{aligned} \tag{6}$$

where $p(y_n|Y_{n-1})$ is obtained by $\int r(y_n|x_n)p(x_n|Y_{n-1})dx_n$.

Smoothing:

$$\begin{aligned} p(x_n|Y_N) &= \int_{-\infty}^{\infty} p(x_n, x_{n+1}|Y_N)dx_{n+1}. \\ &= \int_{-\infty}^{\infty} p(x_{n+1}|Y_N)p(x_n|x_{n+1}, Y_N)dx_{n+1} \\ &= \int_{-\infty}^{\infty} p(x_{n+1}|Y_N)p(x_n|x_{n+1}, Y_n)dx_{n+1} \\ &= p(x_n|Y_n)\int_{-\infty}^{\infty} \frac{p(x_{n+1}|Y_N)p(x_{n+1}|x_n, Y_n)}{p(x_{n+1}|Y_n)}dx_{n+1} \\ &= p(x_n|Y_n)\int_{-\infty}^{\infty} \frac{p(x_{n+1}|Y_N)q(x_{n+1}|x_n)}{p(x_{n+1}|Y_n)}dx_{n+1}. \end{aligned} \tag{7}$$

These formulas (5), (6) and (7) show recursive relation between state distributions. In the linear Gaussian case, the conditional distributions $p(x_n|Y_{n-1})$, $p(x_n|Y_n)$ and $p(x_n|Y_N)$ are characterized by the means and the covariance matrices and (5), (6) and (7) thus are equivalent to the standard Kalman filter and the fixed interval smoothing algorithms. In the general case, however, the conditional distribution of the state $p(x_n|Y_n)$ becomes non-Gaussian and cannot be specified by using only the first two moments. It thus becomes necessary to use a numerical method for the realization of the formulas. This point will be considered in the next section.

The general state space model usually has some unknown parameters. The best values of the parameters can be found by maximizing the log likelihood defined by

$$\begin{aligned} l(\theta) &= \log p(y_1, ..., y_N) \\ &= \sum_{n=1}^{N} \log p(y_n|y_1, ..., y_{n-1}) \\ &= \sum_{n=1}^{N} \log p(y_n|Y_{n-1}). \end{aligned} \tag{8}$$

Here each $p(y_n|Y_{n-1})$ is the quantity appeared in (6).

If we have several candidate models, the goodness of the model can be evaluated by the value of AIC defined by

$$\text{AIC} = -2 \max l(\hat{\theta}) + 2(\text{number of parameters}). \tag{9}$$

Thus the best choice of the model can be made by looking for the one with the smallest value of AIC.

3. Numerical Implementations of the General Smoothing Formulas

In this section, we will show numerical methods for implementing the formulas. For discrete distributions, the implementation is easy. Therefore in this section, we will assume that each distribution has a density function.

3.1 Numerical Approximation to the Densities

In typical situation, the filtering and smoothing formulas can be implemented by using the following operations:

- nonlinear transformation of state
- convolution of two densities
- Bayes formula
- normalization

These operations can be realized by using numerical approximation to the densities. In Kitagawa (1987), each density function was approximated by a continuous piecewise linear (first order spline) function. Here we will show a simple method based on step-function approximation. Each function is expressed by the number of segments, k, location of nodes, x_i, $(i = 0, ..., k)$, and the value of the density at each segment, p_i, $(i = 0, ..., k)$. Specifically, we use the following expressions: $p(x_n|Y_{n-1}) \sim \{k, x_i, p_{ni}\}$, $p(x_n|Y_n) \sim \{k, x_i, f_{ni}\}$, $p(x_n|Y_N) \sim \{k, x_i, s_{ni}\}$, $q(x) \sim \{kq, xq_i, q_i\}$, $r(x) \sim \{kr, xr_i, r_i\}$. We denote these functions by $p_n(x)$, $f_n(x)$, $s_n(x)$, $q(x)$ and $r(x)$, respectively. For simplicity we assume that the nodes are equally spaced and that $\Delta x = x_i - x_{i-1}$.

• Convolution

Consider the convolution of two densities $q(x)$ and

$f_{n-1}(x)$. This can be done by:

$$\begin{aligned} p_{ni} = p_n(x_i) &= \int_{-\infty}^{\infty} q(x_i - y) f_{n-1}(y) dy \qquad (10) \\ &= \sum_{j=1}^{k} \int_{y_{j-1}}^{y_j} q(x_i - y) f_{n-1}(y) dy \\ &= \Delta x \sum_{j=1}^{k} q_{ij} f_{n-1,j}, \end{aligned}$$

here ij satisfies $x_{ij} = x_i - y_j$.

- **Nonlinear Transformation of State**

Assume that the density of x is given as $f(x)$ and that we consider the density, $h(y)$, of $y \equiv g(x)$. If $g(x)$ is a monotone function with an inverse, then $h(y)$ is obtained by $f(g^{-1}(x_n))\frac{dg^{-1}}{dx_n}$. But in general, $h(y) = \{k, x_i, h_i\}$ can be evaluated numerically by the following algorithm:

For $i = 1$ to k

- $y_0 = \min\{g(x_{i-1}), g(x_i)\}$
- $y_3 = \max\{g(x_{i-1}), g(x_i)\}$
- $i_0 = \left[\frac{y_0 - x_0}{\Delta x}\right], i_1 = \left[\frac{y_3 - x_0}{\Delta x}\right] + 1$
- for $j = i_0 + 1$ to i_1
  - $y_1 = \max\{y_0, x_{i_0} + (j-1)\Delta x\}$
  - $y_2 = \min\{y_3, x_{i_0} + j\Delta x\}$
  - $h_j = h_j + \frac{y_2 - y_1}{y_3 - y_0} f_i$

- **Bayes formula**

Given $r(x)$ and the predictive density $p_n(x)$, f_{ni} $(i = 0, 1, ..., k)$ is obtained by

$$f_{ni} = f_n(x_i) = \frac{p_n(x_i) r(y - h(x_i))}{C} = \frac{p_{ni} r_{yi}}{C}. \qquad (11)$$

Here y is the given observation at that stage and $r_{yi} = r(y - h(x_i))$ can be evaluated directly from the function $r(w)$. In (11), C is the normalizing constant given below.

- **Normalization**

$$\begin{aligned} C &= \int_{-\infty}^{\infty} p_n(x) r(y - h(x)) dx \\ &= \sum_{i=1}^{k} \int_{x_{i-1}}^{x_i} p_n(x) r(y - h(x)) dx \qquad (12) \\ &= \Delta x \sum_{i=1}^{k} p_{ni} r_{yi}. \end{aligned}$$

This normalizing constant C can be used for the computation of likelihood given in (9).

3.2 Implementation by FFT

In the above implementation, the most of the computing time is spent for the convolution appeared in the prediction formula. This computation can be significantly reduced by using FFT algorithm based on the following diagram:

$$\begin{array}{llll} f(x), & q(x) & \longrightarrow & p(x) = \int q(y) f(x-y) dy \\ & \Big\downarrow F & & \Big\uparrow F^{-1} \\ F(\omega), & Q(\omega) & \longrightarrow & P(\omega) = F(\omega) Q(\omega) \end{array} \qquad (13)$$

3.3 Gaussian Sum Approximation

In the case of linear state space model with densities, another way of implementing the non-Gaussian fileter is to use Gaussian sum (mixture) approximations to the densities. In this method each density is approximated by a Gaussian sum:

$$\begin{aligned} p(x_n|Y_{n-1}) &= \sum_{i=1}^{m_q} \alpha_i \varphi_i(x_n|x_{n-1}) \\ p(y_n|x_n) &= \sum_{j=1}^{m_r} \beta_j \varphi_j(y_n|x_n) \\ p(x_n|Y_{n-1}) &= \sum_{k=1}^{m_{pn}} \gamma_{kn} \varphi_k(x_n|Y_{n-1}) \qquad (14) \\ p(x_n|Y_n) &= \sum_{l=1}^{m_{fn}} \delta_{ln} \varphi_l(x_n|Y_n) \end{aligned}$$

(15)

where each φ_i is a Gaussian density with appropriate mean and covariance matrix.

Using this approximation, the formulas for prediction and filtering are obtained as follows:

- **Prediction**

$$\begin{aligned} & p(x_n|Y_{n-1}) \\ &= \sum_{i=1}^{m_q} \sum_{k=1}^{m_{fn-1}} \alpha_i \delta_{l,n-1} \int \varphi_i(x_n|x_{n-1}) \varphi_l(x_{n-1}|Y_{n-1}) dx_{n-1} \\ &= \sum_{i=1}^{m_q} \sum_{k=1}^{m_{fn-1}} \alpha_i \delta_{ln-1} \varphi_{il}(x_n|Y_{n-1}) \\ &\equiv \sum_{k=1}^{k_{pn}} \gamma_k \varphi_k(x_n|Y_{n-1}) \qquad (16) \end{aligned}$$

- **Filtering**

$$\begin{aligned} p(x_n|Y_n) &\propto \sum\sum \beta_j \delta_{kn} \varphi_j(y_m|x_n) \varphi_k(x_m|Y_{n-1}) \\ &= \sum_{j=1}^{m_r} \sum_{k=1}^{m_{pn}} \beta_j \gamma_{kn} \varphi_{jk}(y_n|Y_{n-1}) \varphi_{jk}(x_n|Y_{n-1}) \\ &\equiv \sum_{l=1}^{k_{fn}} \delta_l \varphi_l(x_n|Y_n) \qquad (17) \end{aligned}$$

Here $\equiv$ means the reordering, $\gamma_k = \alpha_i \delta_l$ (for some k), $\delta_l = \beta_j \gamma_k \varphi_{jk}(y_n|Y_{n-1})$ (for some l) and φ_{il} and φ_{jk} are obtained by the Kalman filter.

Technical difficulties with this method are as follows:

- The number of necessary Gaussian terms increases very rapidly to infinity, e.g., $m_{fn} = m_{fo} \cdot (m_r \cdot m_r)^n$.

- The smoothing formula cannot be directly realized by this method

The first difficulty can be mitigated by reducing the number of Gaussian components at each step of filtering. Approximate fixed interval smoother can be obtained by fixed lag smoothing which is simply realized by augmenting the state vector.

4. Numerical Examples

We will show various applications of general state space modeling and the general smoothing. The first example of 4.1 and 4.4 are taken from Kitagawa (1987).

4.1 Estimation of Mean Value Function

We consider the estimation of the mean value function of the data. We use a simple first order trend model

$$\begin{aligned} \Delta t_n &= v_n \\ y_n &= t_n + w_n. \end{aligned} \tag{18}$$

Here Δ is the difference operator defined by $\Delta t_n = t_n - t_{n-1}$ and v_n and w_n are white noise sequences that are not necessarily Gaussian. We consider the following model class:

$$\begin{aligned} Model(b): q(v_n) &= C(\tau^2 + v_n^2)^{-b} \\ r(w_n) &= (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{w_n^2}{2\sigma^2}\right\}, \end{aligned} \tag{19}$$

where

$$C = \frac{\tau^{2b-1}\Gamma(b)}{\Gamma(0.5)}. \tag{20}$$

The maximum likelihood estimates of τ^2 and σ^2 for the Gaussian model, Model(∞) were τ^2=0.0122, σ^2=1.043 and the AIC of the model was 1503.03. Fig.1.1 shows the posterior densities of t_n obtained by the Gaussian model. It can be seen that the posterior mean is drifting with the time and does not reflect the jump of the mean value.

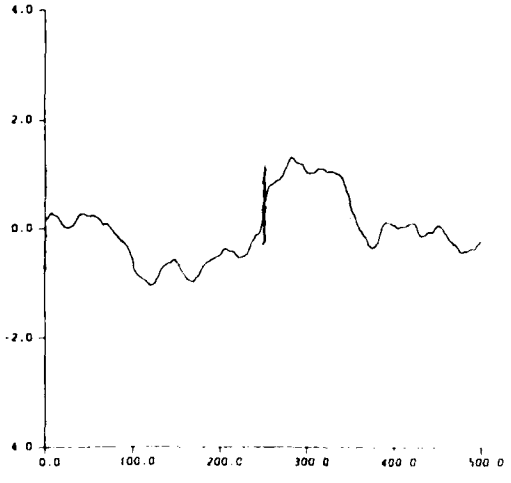


Fig.1.1 Posterior density of t_n obtained by the Gaussian model. (Kitagawa 1987)

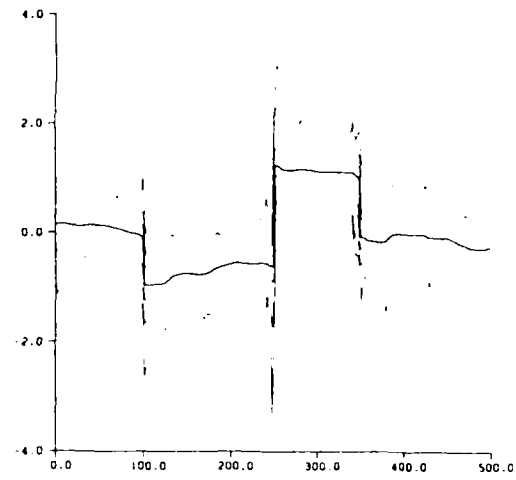


Fig.1.2 Posterior density of t_n obtained by the non-Gaussian model. (Kitagawa 1987)

Among the class of non-Gaussian models, the minimum of AIC (1487.89) was attained at b=0.75, τ^2 = 2.2×10^{-7}, σ^2=1.022. Fig.1.2 shows the posterior density of t_n obtained by this non Gaussian model. Comparing with Fig.1.1, it becomes clear that the non Gaussian model, Model(0.75) has better ability to reproduce the jump of the mean level automatically.

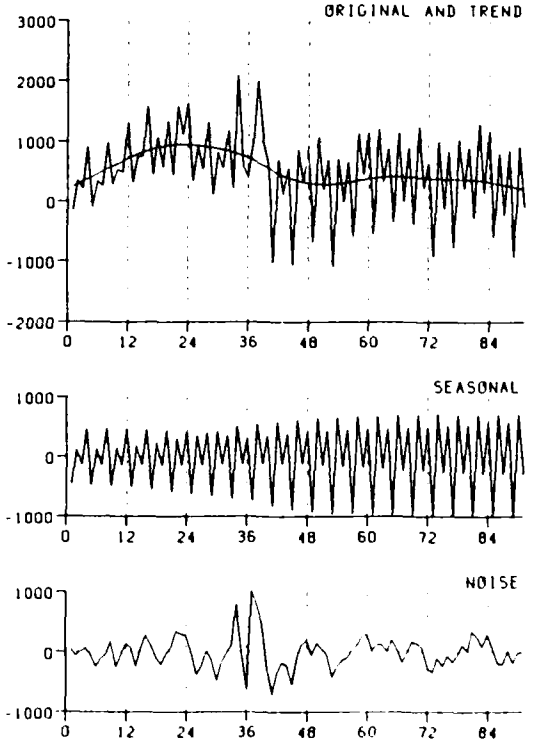


Fig.1.3 Seasonal data, estimated trend, seasonal and noise components by a Gaussian model

•Non-Gaussian seasonal adjustment

This method of trend esstimation can be extended to seasonal adjustment. The state space model for the seasonal adjustment is given in Kitagawa and Gersch (1984). But here neither sytem noise v_n nor observational noise w_n are assumed to be Gaussian. Since the state dimension of the seasonal adjustment model is large, we used a Gaussian sum approximation. Fig.1.3 shows the quarterly record of the increase of inventories of private companies in Japan (1965 to 1983). Also shown are the trend, seasonal and the noise components estimated by a Gaussian model. The estimated trend is too smooth and the seasonal component changes gradually. On the other hand, Fig.1.4 shows the results by non-Gaussian model. We can see that the trend jumps up and down arround at 1973-1974 when the oil crisis took place. The seasonal pattern also changed significantly during this time period.

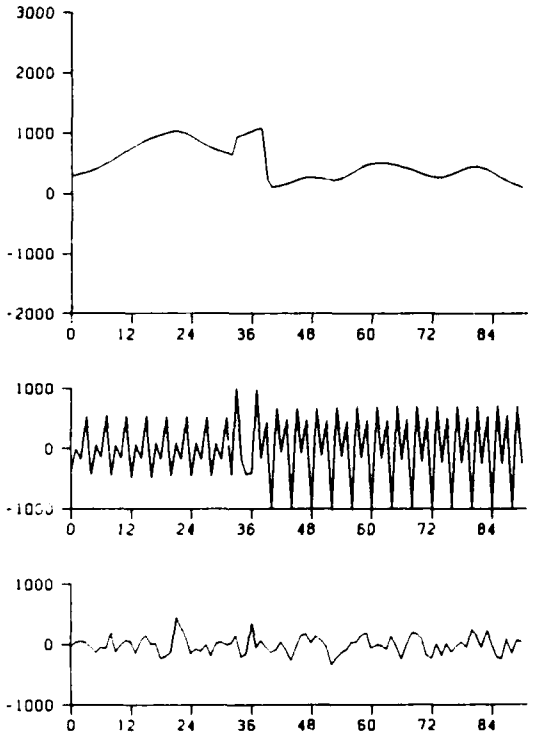


Fig 1.4 Estimated trend, seasonal and noise components by a non Gaussian model.

4.2 Estimation of Changing Variance

We consider the estimation of changing variance of a nonstationary time series. Assume that the time series $y_1, \ldots, y_N$ is an independent Gaussian sequence with time-varying variance r_m. Then the transformed series u_m defined by

$$u_m = \log(y_{m-1}^2 + y_m^2) \qquad (21)$$

has the property that $u_m - \log r_m$ is an independent random variable distributed as the logarithm of an exponential distribution. Therefore, we can estimate the log-variance by using the model

$$\begin{aligned} \Delta^k t_m &= v_m \\ u_m &= t_m + w_m, \end{aligned} \qquad (22)$$

with $r(w) = exp\{w - e^w\}$.

•Smoothing periodogram

Since the periodogram is distributed as an exponential distributuion, as a natural application of this method, we can smooth the log-periodogram by the above method. In spline smoothing, Wahba(1980) approximated the density $r(w)$ by a Gaussian distribution with the same first and second moments as those of $r(w)$. But here by our method, we can smooth the series without using Gaussian approximation. Fig.2 shows the log-periodogram and the smoothed log-periodogram obtained by the AIC best

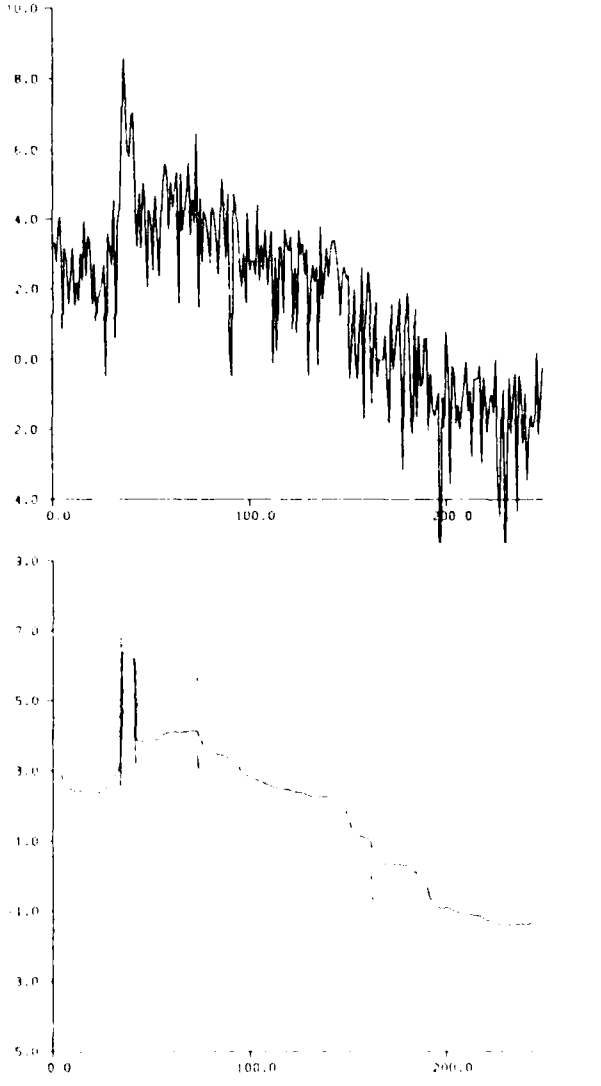


Fig 2 Log periodogram and smoothed log-periodogram by a non Gaussian model

model, e.g.,the second order trend model ($k = 2$) with Cauchy system noise input.

4.3 Estimation of Changing Spectrum

It is well known that the coefficients of an AR model can be estimated recursively by obtaining partial autocorrelation coefficients. We extend this method to nonstationary AR model

$$y_n = \sum_{i=1}^{K} a_{in} y_{n-i} + w_n. \tag{23}$$

The models for smoothing time-varying partial autocorrelation coefficients are

$$\begin{cases} f_n^{(k-1)} = a_{kn} b_{n-k}^{(k-1)} + f_n^{(k)} \\ \Delta a_{kn} = v_n \end{cases}$$
$$\begin{cases} b_{n-k}^{(k-1)} = c_{kn} f_n^{(k-1)} + b_n^{(k)} \\ \Delta c_{kn} = u_n \end{cases} \tag{24}$$

with $f_n^{(0)} = b_n^{(0)} = y_n$. $f_n^{(k)}$ and $b_n^{(k)}$ are respectively forward and backward prediction error of the autoregressive model of order k. In stationary case, $v_n = u_n = 0$ and $a_{kn} = c_{kn}$ are identical to the partial autocorrelation coefficients. The distribution of the noise inputs may be either Gaussian or non-Gaussian. After estimating time-varying AR coefficients by the smoothing method, we can estimate the instantaneous spectrum of nonstationary process by

$$p_n(f) = \frac{\sigma^2}{|1 + \sum a_{jn} \exp(2\pi i j f)|^2}. \tag{25}$$

Fig.3 shows the estimated changing spectrum of a seismic data. The AR coefficients are estiamted by assuming that

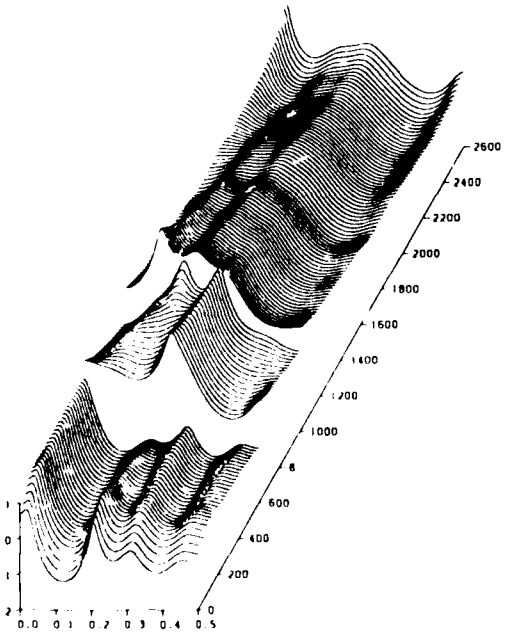


Fig.3 Estimated changing spectrum of a seimic data

$f_n^{(k)}$ and $b_n^{(k)}$ are Cauchy and w_n is Gaussian. We can see that the arrival of P and S waves are clearly detected by this method.

4.4 Inhomogeneous Discrete Process

The general state space model can be applied to the estimation of time-varying mean of the discrete distributions. We consider the number of rainy days over 1mm in Tokyo for each day during 1983-84. The problem is to estimate the probability, p_n, of occurence of rainfall on a specific calendar day which is believed to be gradually changing with time.

We estimated the probability of rainfall by the following model:

$$\Delta^k v_n = v_n$$
$$z_{l_n}(m_n|q_n) = \binom{l_n}{m_n} p_n^{m_n}(1-p_n)^{l_n-m_n}. \tag{26}$$

Here $q_n = \log\{p_n/(1-p_n)\}$, l_n is the number of observations at the n-th day, m_n the number of rainy days and p_n the time dependent mean of the binomial distribution. The estiamted rainfall probability for Tokyo obtained by this method is shown in Fig.4.

•Nonstationary Poisson Process

The same method can be used for the estimation of time-varying mean of the Poisson distribution. This problem occures in the analysis of X-ray stars. The main problem is the analysis of quasi-periodicity of the time fluctuation of the Poisson mean observed at a satellite. But for some stars, the time constant is very short (e.g. mili-second), and the signal (change of mean) is only a few percent of the mean level (and the variance)and the series looks like the one with very low signal to noise ratio. For such a series, this method can be used to extract the signal.

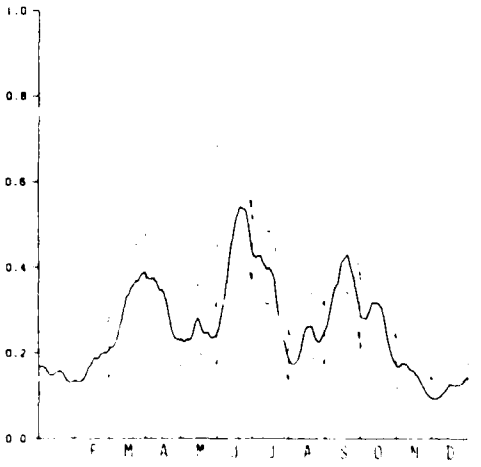


Fig.4 Estimated rainfall probability for Tokyo. (Kitagawa 1987)

4.5 Quasi Periodic Process

The famous Wolf sunspot number data exhibits the approximate repetition of a pattern but both the period and the amplitude are not so definite and change gradually. This type of phenomena can be seen in ecological data (e.g., the Canadian lynx data) and many varieties of air polution data. Although such series are frequently modeled by AR, ARMA or AR plus sumisoidals models, none of those models seems quite satisfactory for the prediction with more than one lead time. For a time series with quasi periodic character, by using a model

$$\begin{aligned} \Delta^k t_n &= c_n \\ \Delta^k c_n &= a_n \\ y_n &= c_n \left\{ \sum_{j=0}^{m} a_j \sin t_n + \sum_{j=1}^{m} b_j \cos t_n \right\} + w_n, \end{aligned} \tag{27}$$

we can estimate phase and amplitude of the model. Fig.5.1 shows the sunspot number data, the estimated phase, log-amplitude and the cyclic function of the model

To check the prediction ability, we consider a simple model with constant amplitude:

$$\begin{aligned} y_n &= \cos(\theta_n) + w_n \\ \theta_n &= \theta_{n-1} + 0.2\pi(1 + 0.01 t_n) \\ t_n &= 1.85 t_{n-1} - 0.855 t_{n-2} + c_n \\ w_n &\sim N(0, 0.04) \\ c_n &\sim N(0, 1) \end{aligned} \tag{28}$$

160 observations was generated with this model. The parameter of the model and the state are estimated by using the first 120 observations. The second order trend model was the AIC best. Fig.5.2 shows the increasing horizon prediction of the state x_n (or θ_n) and of the observation y_n obtained by

$$\begin{aligned} p(x_{n+k}|Y_n) &= \int p(x_{n+k}|x_{n+k-1}) p(x_{n+k-1}|Y_n) dx_{n+k-1} \\ p(y_{n+k}|Y_n) &= \int p(y_{n+k}|x_{n+k}) p(x_{n+k}|Y_n) dx_{n+k} \end{aligned} \tag{29}$$

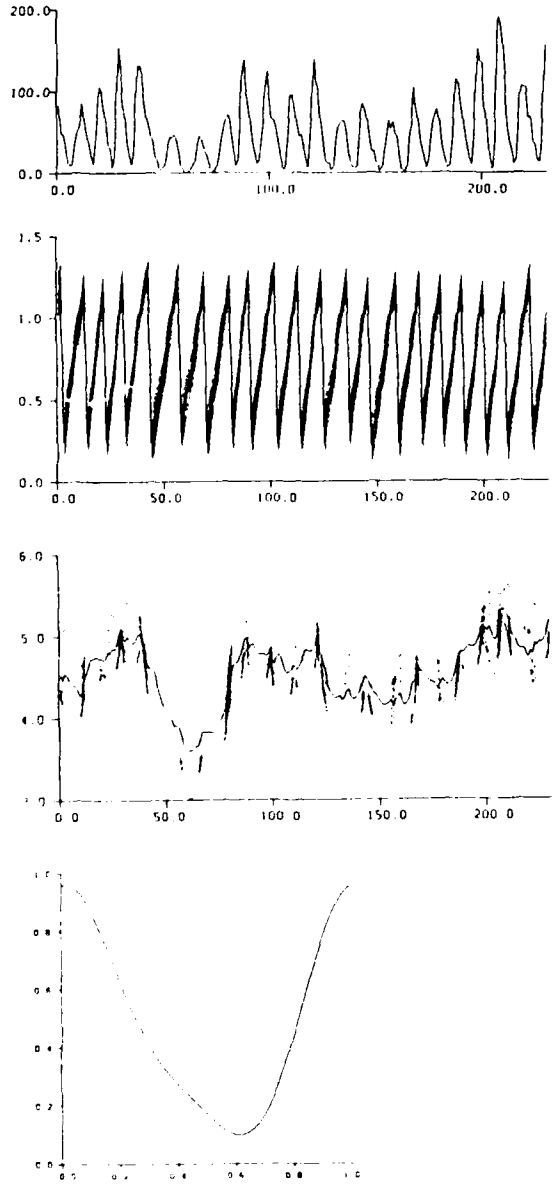


Fig.5.1 Sunspot number data, estimated phase, amplitude and cyclic function of the model.

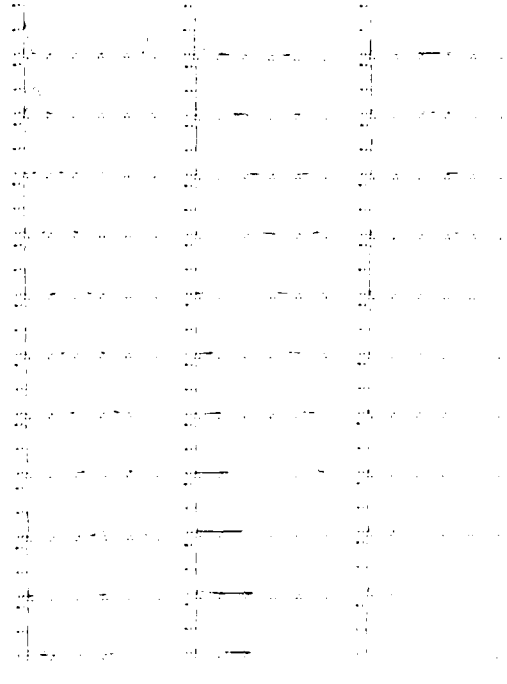


Fig.5.2 Predictive density of $p(x_n|Y_{120})$ $(n = 121, 153)$

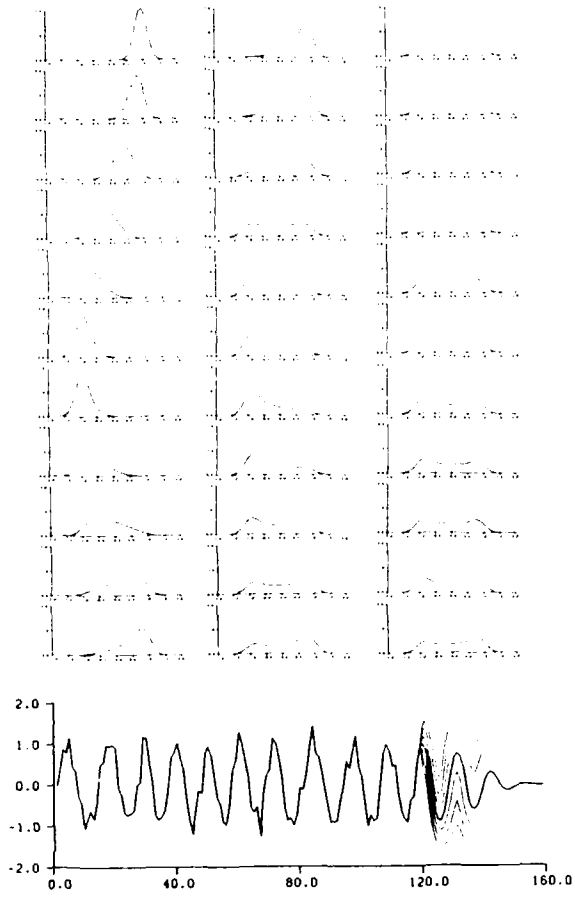


Fig.5.3 Predictive density of $p(x_n|Y_{120})$.

4.6 Nonlinear smoothing

We consider the data artificially generated by the following model which was originally used by Andrade Netto et. al.(1978) and mentioned in the rejoinder of Kitagawa (1987):

$$\begin{aligned} x_n &= \frac{1}{2}x_{n-1} + \frac{25x_{n-1}}{1+x_{n-1}^2} + 8\cos(1.2n) + v_n \\ y_n &= \frac{x_n^2}{20} + w_n \end{aligned} \tag{30}$$

The data is shown in Fig.6.1. The problem is to estimate the true signal x_n from the sequence of observations $\{y_n\}$ assuming that the model (30) is known. Our nonlinear filter and smoother were applied to this problem. For comparison, the well-known extended Kalman filter was also applied. Fig.6.2 shows the posterior densities $p(x_n|Y_m)$, $m = 16, \ldots, 20$ and 100. The densities shown in the left hand side of the figure are obtained by the extended Kalman filter and the linearlized fixed interval smoother (Sage and Mersa 1971). The ones in the right hand side of the figure show the results by our nonlinear filter and smoother. These figures show a typical situation where these two algorithms yield quite different results.

Fig.6.3 shows the posterior density $p(x_n|Y_n)$ obtained by the extended Kalman filter based linearized smoother. It can be seen that the estimates have the tendency of divergence and does not satisfactorily reproduced the true signal shown Fig.6.1. On the other hand, Fig.6.4 shows the results by our nonlinear smoother. We can see that remarkably good results was obtained by our smoother.

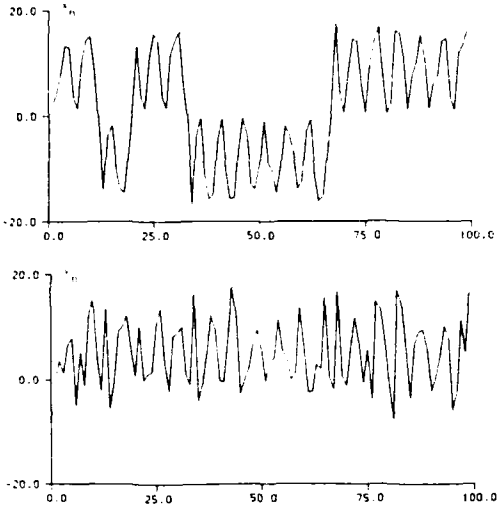


Fig.6.1 Simulated x_n which is assumed to be unkown and the observations y_n.

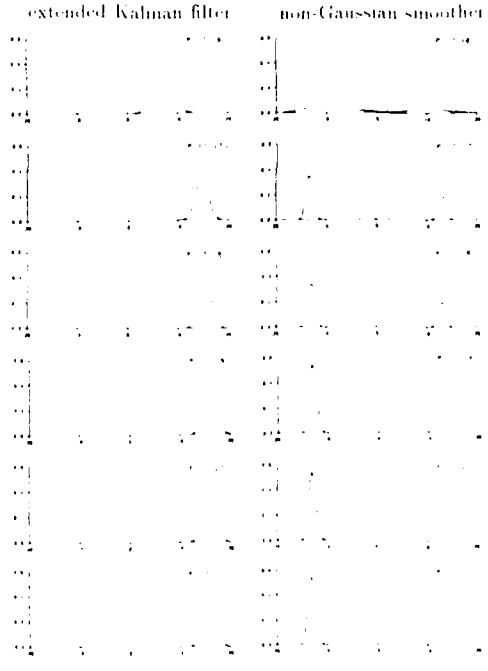


Fig.6.2 Posterior densities of $p(x_n|Y_m)$, $m = 16, \ldots, 20$ and 100 obtained by extended Kalman filter and non-Gaussian smoother.

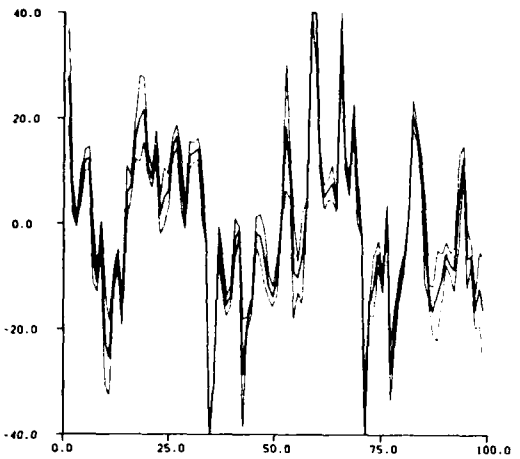


Fig.6.3 Posterior density of $p(x_n|Y_N)$ obtained by the extended Kalman filter based smoother.

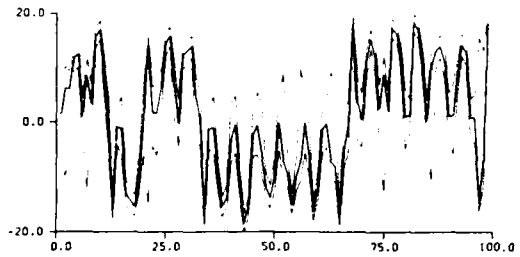


Fig.6.4 Posterior density of $p(x_n|Y_N)$ obtained by the non-Gaussian smoother.

The second example of nonlinear smoothing is a passive receiver problem. A similar problem was considered by Bucy and Senne(1970). In this example, the target is gradually moving on the two dimensional space. Fig.6.5 shows an example of this trajectory. This target is observed by the scalar nonlinear measurement function

$$y_n = h(x_n^1, x_n^2) + w_n \tag{31}$$

where

$$\begin{aligned} h(x_n^1, x_n^2) &= \tan^{-1}\left\{\frac{x_n^2 - \sin\beta_n}{x_n^1 - \cos\beta_n}\right\} \\ \beta_n &= \beta_{n-1} + \Delta\beta \end{aligned} \tag{32}$$

Here β_0 and $\Delta\beta$ are given constants and w_n is a Gaussian white noise with known variance σ^2. This is a simple example of vector tracking problem of a moving object by observing the relative angle observed on a rotating observatory. Fig.6.6 shows an example of y_n which is generated by the model(31). For the estimation of this moving object, we consider the following simple smoothness prior model

$$\begin{aligned} \Delta^k x_n^1 &= v_n^1 \\ \Delta^k x_n^2 &= v_n^2 \end{aligned} \tag{33}$$

Here v_n^1 and v_n^2 are mutually independent Gaussian white noise sequence with variances, τ_1^2 and τ_2^2, respectively. The smoothness prior model (33) with the observation model (31) constitute our nonlinear state space model for estimating the location of the object. It should be noted that the Gaussianity of neither v_n nor w_n are necesssary in our model. The value of τ_1^2 and τ_2^2 are unknown but can be estimated by maximizing the log-likelihood defined by (8). Fig.6.7 shows the contour of the posterior density $p(x_n^1, x_n^2|Y_N)$ for n=20,40,60,and 80.

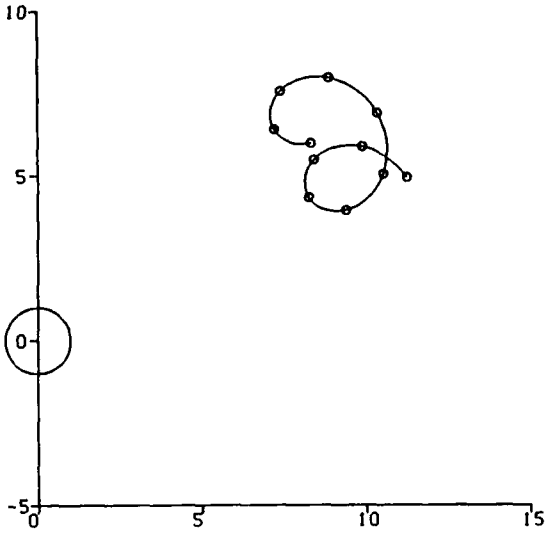


Fig.6.5 Trajectory of (x_n^1, x_n^2).

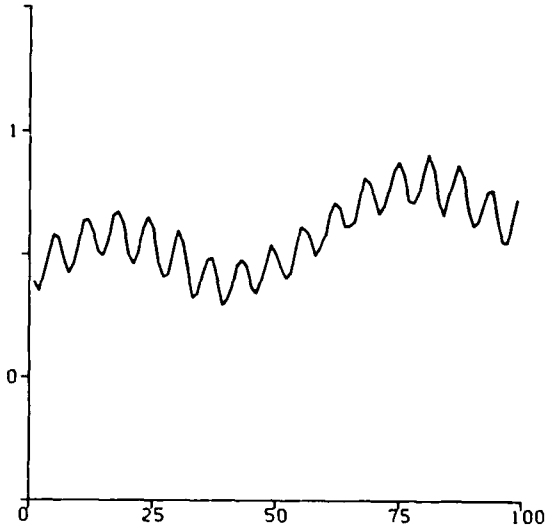


Fig.6.6 Observed y_n.

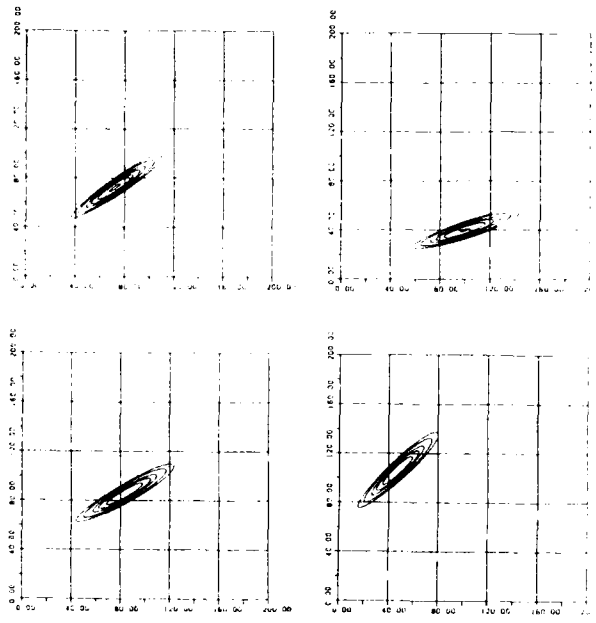


Fig.6.7 Contour of posterior densities $p(x_n^1, x_n^2|Y_N)$ for n=20,40,60 and 80.

6. CONCLUDING REMARKS

By the use of general non-Gaussian *state space model*, we can treat various types of time series. Recursive filtering and smoothing formulas can be derived for this generic state space model. The direct numerical method for non-Gaussian filtering and smoothing is practical at least for lower dimensional problem.

For higher order system, however, it involves intensive computations. The most significant part of the computing time is spent for the convolution. The amount of the computation for convolution is roughly the order of $k^{m+1}n$, here k is the number of segments, m is the state dimension and n is the data length. For rough idea, the CPU time spent for examples in section 4.1 ($m = 1$) and 4.6 ($m = 2$ and 4) are 6 second and 450 second, respectively, by a 14 MIPS computer. Obviously for higher order systems, we need more additional effort. That will include the use of supercomputer, development of special algoerithm or hardware for fast convolution, integration or FFT and reduction of the number of necessary operations based on spline or Gaussian sum approximations.

In our model, it is assumed that the noise distribution is nonsigular and the conditional distribution $p(x_n|x_{n-1})$ is well-defined. Ansley and Kohn (1987) pointed out that this does not the case for some important problem. Necessary modification for such a case is shown in that article

REFERENCES

Akaike, H. (1973), 'Information Theory and an Extension of the Maximum Likelihood Principle', in Second International Symposium on Information Theory, B.N. Petrov and F.Caski eds, Budapest, Akademiai Kiado, 267-281.

Akaike, H. (1980), 'Likelihood and Bayes Procedure,' in Bayesian Statistics,' J.M.Bernardo, M.H. De Groot, D.V. Lindley and A.F.M. Smith, eds., University Press Valencia, Spain, 143-166.

Anderson, B.D.O. and Moore, J.B. (1979), Optimal Filtering, New Jersey, Prentice-Hall.

Andrade Netto, M.L., Gimeno,L. and Mendes, M.J. (1978), 'On the Optimal and Suboptimal Nonlinear Filtering Problem for Discrete-Time Systems,' *IEEE Transactions on Automatic Control*, AC-23 No.6,1062-1067.

Ansley, C.F. and Kohn, R. (1987) Discussion of the paper by Kitagawa, *Journal of American Statistical Association*, Vol.76, No.400

Bucy, R.S. and Senne, K.D. (1971), 'Digital Synthesis of nonlinear filters', *Automatica*, 7, 287-289.

de Figueiredo, R.J.P. and Jan, Y.G. (1971), 'Spline Filters', Proceedings of the 2nd Symposium on Nonlinear Estimation Theory and its applications, San Diego, 127-141.

Kitagawa, G. (1983), 'Changing Spectrum Estimation', Journal of Sound and Vibration, 89, No.4, 433-445.

Kitagawa, G. and Gersch, W.(1984), 'A Smoothness Priors-State Space Approach to the Modeling of Time Series with Trend and Seasonality,' *Journal of the American Statistical Association*, 79, No.386, 378-389

Kitagawa, G. (1986), Non-Gaussian Smoothness Prior Approach to Irregular Time Series Analysis, Adaptive Systems in Control and Signal Processing 1986, eds. K.J. Aström and B. Wittenmark, Pergamon Press, Oxford.

Kitagawa, G. (1987), 'Non-Gaussian state Space Modeling of Nonstationary Time Series,' *Journal of American Statistical Association*, Vol.76, No.400 1032-1063.

Sage, A. P. and Mersa, J. L. (1971), 'Estimation Theory with Applications to Communications and Control', McGraw-Hill Series in System Science, McGraw-Hill, New York.

Shiller, R. (1973), 'A Distributed Lag Estimator Derived From Smoothness Priors', *Econometrica*, 41, 775-788.

Sorenson, H.W. and Alspach, D.L. (1971), 'Recursive Bayesian Estimation Using Gaussian Sums,' *Automatica*, 7, 465-479.

Whittaker, E.T. (1923), 'On a New Method of Graduation', Proceedings of the Edinbrgh Mathematical Soceity, 41, 81-89.

INTERIOR POINT METHODS FOR LINEAR PROGRAMMING PROBLEMS[1]

P.T. Boggs, P.D. Domich, J.R. Donaldson and C. Witzgall, National Bureau of Standards

1. Introduction

Interior point methods for linear programming problems are certainly not new. Many people have been intrigued by the notion of going through the polytope rather than proceeding from vertex to vertex around the polytope as required by the simplex method. This idea received fresh impetus from the startling anouncements of Karmarkar [Kar84] who claimed that his new interior point method based on projective methods solved a certain large linear programming problem 50 times faster than the simplex method. Furthermore, this algorithm was provably polynomial in the number of operations required. Thus for the first time, there was a polynomial algorithm for linear programming that actually held the promise of outperforming the simplex method. Since then, there have been many studies related to Karmarkar's method and renewed interest in other interior point methods such as the barrier function method and Huard's method of centers.

In this paper we present computationally efficient interior point methods based on Huard's method of centers. We confirm the need to keep iterates close to the center trajectory of the polytope in order to get good performance, and we derive and present numerical results for two specific *multi-directional* procedures. The best of our procedures is based on solving a two-dimensional linear programming problem at each step. This method compares very favorably with other recent interior point procedures reported in the literature.

In the presentation that follows, we take the linear programming problem to be in the form

$$\begin{aligned} &\min_u c^T u \\ &\text{subject to } Au \le b \end{aligned} \tag{1.1}$$

where $c, u \in \mathbf{R}^n$, $A \in \mathbf{R}^{m \times n}$, and $b \in \mathbf{R}^m$. Although we assume that the problem is bounded and that A has full column rank, it is not necessary to assume that the constraints have a full dimensional interior since the **big-M** procedure used here to find an initial feasible point will always have one. In that case, the Phase I solution will be the optimal solution.

The remainder of this paper is organized as follows. In §2 we give a description of Huard's original method of centers, and we consider some generalizations. In particular, we show that smooth trajectories exist that connect any initial feasible point to an optimal solution. Some of these trajectories, however, get arbitrarily close to an exponential number of vertices, and we argue that recentering is therefore desirable. In §3 we derive two specific algorithms that incorporate a recentering strategy. Finally, in §4 we give some numerical results that show the promise of our approach. The details of the methods and results presented here are contained in [BDDW88]; additional theoretical developments are in [WBD88].

2. The Method of Centers

In this section we describe the method of centers and show how to obtain a smooth trajectory rather than a sequence of points. We generalize these results by posing an initial value problem in ordinary differential equations whose solutions are trajectories that connect any feasible initial point to an optimal solution. The problem of trajectories that get arbitrarily close to an exponential number of vertices is then examined. We overcome this problem without sacrificing the essential properties of the original system by deriving a modified differential equation that has a recentering component.

The notation required to describe the method of centers is defined as follows. Let the set of residuals corresponding to the constraints of (1.1) be

$$r_k(u) = b_k - A^T_k, \qquad k = 1, \ldots, m.$$

Note that if $r_k(u) \ge 0$, $k = 1, \ldots, m$, then u is a feasible point. Next, define a residual corresponding to the objective function

$$r_0(u,t) = t - c^T u.$$

Here t is a scalar variable that is meant to correspond to a previous value of the objective function. In particular, let u_0 be a feasible point and let

$$t_0 = c^T u_0.$$

Then if $r_0(u, t_0) > 0$, u yields a lower objective function value than u_0, i.e., $c^T u < c^T u_0$.

The *center* of the polytope defined by the constraints of (1.1) and the objective constraint for $t = t_0$ is the feasible point, u_1, that solves

$$\max_u \log \left[r_0(u, t_0) \prod_{k=1}^{m} r_k(u) \right]$$

[1]Contribution of the National Bureau of Standards and not subject to copyright in the United States.
This research was supported in part by ONR Contract N-0014-87 F0053.

or

$$\max_u \left[\log r_0(u, t_0) + \sum_{k=1}^{m} \log r_k(u) \right].$$

Now set

$$t_1 = c^T u_1,$$

and define u_2 as the solution to

$$\max_u \left[\log r_0(u, t_1) + \sum_{k=1}^{m} \log r_k(u) \right].$$

Continuing this process, a sequence of iterates $\{u_i\}$ is obtained. It can be shown that $\{u_i\}$ converges to an optimal solution as $i \to \infty$. This procedure is Huard's original method of centers [Hua67] applied to the linear programming problem. An implementation of this method was shown by Renegar [Ren86] to possess an equivalent polynomial complexity bound to that of Karmarkar's original method [Kar84].

It is easy to see that by continuously moving the constraint corresponding to the objective function, one obtains a continuous trajectory rather than a set of points. Every point on that trajectory can be viewed as a function of t. Specifically, let

$$L(u, t) = \log r_0(u, t) + \sum_{k=1}^{m} \log r_k(u).$$

Then for any value of t, $u(t)$ satisfies

$$\nabla_u L(u(t), t) = 0. \tag{2.1}$$

By differentiating (2.1) with respect to t, an expression is obtained for the change in $u(t)$ as a function of t, i.e.,

$$\nabla_{uu} L(u, t) u'(t) + \nabla_{ut} L(u, t) = 0,$$

or

$$u'(t) = -\nabla_{uu} L(u, t)^{-1} \nabla_{ut} L(u, t). \tag{2.2}$$

While the above differential equation characterizes the trajectory, an initial condition needs to be supplied to complete the specification. We consider therefore

$$\begin{aligned} u'(t) &= \nabla_{uu} L(u, t)^{-1} \nabla_{ut} L(u, t) \\ u(t_0) &= u \\ t_0 &= c^T u + \epsilon \\ \epsilon &> 0. \end{aligned} \tag{2.3}$$

By taking $u = u_1$ from above and $\epsilon = c^T u_0 - c^T u$ we obtain the desired trajectory. It is of interest, however, to consider any initial feasible point u and any $\epsilon > 0$, and to assess the solution to (2.3). In general, for any such u and ϵ,

$$g = \nabla_u L(u, t_0)$$

is not equal to zero. Thus, if we require that along a trajectory $u_g(t)$,

$$\nabla_u L(u_g(t), t) = g,$$

then $u_g'(t)$ satisfies (2.2) and hence the initial value problem (2.3). When $g = 0$ the resulting trajectory is referred to as the *center trajectory* while if $g \neq 0$ the trajectory is called an *off-center trajectory*. The theoretical properties of these trajectories are contained in [WBD88].

Computing the actual derivatives of L and substituting these in (2.2) yields

$$u'(t) = \left(A^T D^2 A + \frac{cc^T}{(t - c^T u)^2} \right)^{-1} \frac{c}{(t - c^T u)^2}, \tag{2.4}$$

where D is defined as

$$D = \operatorname{diag}\left\{ \frac{1}{r_k},\ k = 1, \ldots, m \right\}.$$

For

$$X = \operatorname{diag}\left\{ \frac{(t - c^T u)^2}{r_k},\ k = 1, \ldots, m \right\},$$

(2.4) can be rewritten as

$$u'(t) = \left(A^T X A + cc^T \right)^{-1} c. \tag{2.5}$$

Applying the Sherman-Morrison-Woodbury formula to the matrix in (2.5) results in

$$u'(t) = \eta \left(A^T X A \right)^{-1} c \tag{2.6}$$

where η is a scalar. A numerical procedure for solving linear programming problems can then be obtained by numerically integrating (2.5) or (2.6). The use of Euler's method, for example, yields a direction that is known as the dual affine direction [ARV86].

In Witzgall *et al.* [WBD88], we show that all trajectories converge to a single optimal solution, even when the optimal solutions are not unique. In the case of a single optimal solution at a vertex, all of the trajectories converge to that vertex, and the tangents of these trajectories similarly converge.

As we discuss in [BDDW88], there are paths that stay arbitrarily close to the boundary of the polytope (see also [MS86]). This is corroborated by the fact that $\nabla_u L(u(t), t)$ stays constant; if it is large initially, which it will be if u_0 is close to the boundary, then it must stay large thereafter. Thus the dual affine direction can produce *long paths* [MS86] in the polytope, i.e., paths that visit an exponential number of vertices. These paths mirror the long paths exhibited by the simplex method, and it follows that poor performance is possible.

Recall however that the center trajectory is defined by the condition that $L(u, t)$ be maximized as a function of u. The natural method of solving this optimization problem is Newton's method where the step is given by

$$\nabla_{uu} L(u, t)^{-1} \nabla_u L(u, t). \tag{2.7}$$

We can incorporate this direction into the differential equation (2.2) to obtain

$$u'(t) = -\nabla_{uu}L(u,t)^{-1}[\nabla_{ut}L(u,t) - \phi\nabla_u L(u,t)] \quad (2.8)$$

for an arbitrary positive constant ϕ. The sign appears to be wrong, but recall that the integration is backwards from t_0 to the optimal value $t^* < t_0$.

It would seem that (2.8) would not satisfy any condition such as (2.1). By differentiating

$$\nabla L(u(t),t) = e^{\phi(t-t_0)}g$$

with respect to t, however, and solving for $u'(t)$ as before, we obtain (2.8). Thus, along any solution to (2.8), the value of $\|\nabla_u L\|$ decreases. It does not, however, go to 0 since the upper limit of integration is t^* and not $-\infty$. The amount of recentering correction can also be made to vary with t. If

$$\Phi(t) = \int_{t_0}^{t} \phi(s)ds$$

then deriving the differential equation with respect to t using

$$\nabla_u L(u(t),t) = e^{\Phi(t)}g$$

yields (2.8), but with ϕ replaced by $\phi(t)$.

The theoretical properties of (2.8) are contained in [WBD88]. Note however, that while recentering is intuitively appealing when far from the optimal vertex, it may *actually slow final convergence. In* [BDDW88] *we observe* that a *negative* component in the recentering direction often yields a better direction as the iterates approach the optimal vertex. Further discussion on this point is included in §3, which explores numerical algorithms based on the search directions derived here.

3. New Algorithms

In the previous section we motivated the choice of the dual affine search direction by using Euler's method for ordinary differential equations (ODE). Different search directions could be generated by considering other methods for the numerical integration of initial value problems. Since the precise determination of the actual trajectory from an initial feasible point to the solution is not of interest here, the ODE analysis is used only to suggest search directions; other considerations dictate the distance to travel in those directions.

For example, a typical dual affine procedure uses a steplength that is a large percentage of the distance to the boundary of the polytope and thus does not attempt to follow a single trajectory. One can easily see that this type of algorithm might perform poorly. While the current estimate might be on a "well-behaved" trajectory, the next iterate might be on a trajectory that gets arbitrarily close to an exponential number of vertices of the polytope.

Long paths can be avoided by including a recentering component in the search direction. Such an approach aims at keeping the iterates more interior to the polytope which helps maintain nonsingularity in $A^T D^2 A$. Various strategies for combining a recentering direction with the dual affine direction are possible. Such "multi-direction" methods are discussed in the remainder of this section.

A *multi-direction* method attempts to combine a cost improvement direction, s_c, with a recentering direction, s_r, so that the iterates remain sufficiently close to the center trajectory and thus do not display the convergence problems described above. The value of multi-directional search procedures is well appreciated in the recent work involving interior point methods, e.g., [Kar85]. Closer analysis by Gonzaga [Gon87] of various standard interior point approaches demonstrated that each consisted of two basic directions: a cost improvement direction, and a recentering direction. In this study, we consider three different multi-dimensional search approaches using both the standard search directions and new ones. These approaches are called the composite method, the two-step method, and the two-dimensional subspace method.

The *composite method* is conceptually the least complicated of the multi-direction methods. Using this method, the two component directions are combined at each iteration to form a single direction

$$s = s_c + \phi(t)s_r$$

as suggested by (2.8), where $\phi(t)$ is a weight that determines the contribution of each component to the combined direction. Unfortunately, we have not been able to find a value of $\phi(t)$ that performs consistently well in practice.

In addition, the selection of an appropriate steplength for the composite direction depends heavily on the value of $\phi(t)$ selected. Projecting a search direction heavily dominated by a recentering component to within 99% of the boundary won't necessarily yield an improved trajectory. Likewise, attempting to recenter using a quadratic line search with a direction dominated by the dual affine search direction is not practical since the slope of the quadratic model of $\nabla_u L(t,u)$ will be approximately zero (see §2). Because of these problems, the composite method is not discussed further.

The *two-step method* uses the two component directions independently. At each iteration, a cost improvement step is followed immediately by a recentering step. This eliminates the need to explicitly specify the weight $\phi(t)$ as required by the composite method. The two-step method also eliminates the problem associated with the composite method of selecting an appropriate steplength. Since the steps in the cost improvement and recentering directions are made independently, the steplength of each step can also be specified separately. This method, described further in §3.1, is found to work well in practice.

We derive the *two-dimensional subspace method* by noting that a cost improvement and recentering direction, provided they are not co-linear, define a two-dimensional cross section of the polytope. Because of the reduced dimension, the cost function can be easily minimized on this two-dimensional section. The solution to the reduced problem then defines a search direction that combines the original two directions. The two-dimensional subspace method is described in detail in §3.2. As reported in §4, this method also performs well.

Each of these multi-direction methods include derivatives of L that contain the term $t - c^T u$ and, implicitly, its initial value, ϵ. Since no effort is being made to remain on a particular trajectory, it is reasonable to "start over" at each step, i.e., to pick a new value of ϵ at each step and to ignore t. Thus, in the following, the derivatives are written using ϵ and not $t - c^T u$. For example,

$$\begin{aligned} \nabla_u L(u,t) &= -AR - \frac{c}{t - c^T u} \\ &= -AR - \frac{c}{\epsilon}. \end{aligned}$$

Particular choices for ϵ are discussed below in context.

3.1. Two-Step Methods

As outlined above, the two-step procedure follows a cost improvement step with an independent recentering step. The dual affine direction

$$s_{da} = \left(A^T D^2 A\right)^{-1} c$$

is the obvious choice for the cost improvement direction. There are many possible choices for the recentering direction.

One such choice is the Newton recentering direction defined by (2.7), i.e.,

$$s_n = \left[A^T D^2 A + \frac{cc^T}{\epsilon_n^2}\right]^{-1} \left(A^T R - \frac{c}{\epsilon_n}\right), \qquad (3.1)$$

where

$$\epsilon_n = \frac{-c^T \left(A^T D^2 A\right)^{-1} c}{c^T (A^T D^2 A)^{-1} A^T R}.$$

The choice of ϵ_n, originally suggested by McCormick [McC87], results in a Newton recentering direction that is orthogonal to the direction c. The value of ϵ_n thus minimizes the ℓ_2 norm of the vector

$$\left(A^T D^2 A\right)^{\frac{1}{2}} \left(A^T R + \frac{c}{\epsilon_n}\right).$$

In the presence of ill-conditioning in $A^T D^2 A$, the steepest descent recentering direction

$$s_{sd} = A^T R - \frac{c}{\epsilon_{sd}},$$

where

$$\epsilon_{sd} = \frac{c^T c}{c^T A^T R},$$

might be preferable to the Newton direction. The value ϵ_{sd}, which was originally suggested by Fiacco and McCormick [FM68] in the context of barrier functions, minimizes the ℓ_2 norm of $\nabla_u L(u,t)$. This value produces a steepest descent recentering direction that is orthogonal to the cost direction c, i.e.,

$$c^T \left(A^T R - \frac{c}{\epsilon_{sd}}\right) = c^T \nabla_u L(u,t) = 0.$$

In the simplest implementation of the two-step method, both component directions are computed at the current estimate u_i. The first step taken is some large percentage of the distance to the boundary in the dual affine direction, $\left(A^T D^2 A\right)^{-1}_{u_i} c$. This step results in an intermediate point $\tilde{u}_i$. The trajectory is then corrected using the recentering direction computed at u_i provided that the recentering direction forms a negative inner product with

$$\nabla_u L(\tilde{u}_i, t) = \left(-A^T R - \frac{c}{\epsilon}\right)_{\tilde{u}_i},$$

where ϵ is either ϵ_n or ϵ_{sd} depending on which recentering direction is used. A quadratic model is used to determine the steplength for the recentering direction.

The two-step method is improved if the recentering direction is updated at the intermediate point $\tilde{u}_i$. For the steepest descent direction, this simply means computing

$$s_{sd} = \left(A^T R - \frac{c}{\epsilon_{sd}}\right)_{\tilde{u}_i}.$$

For the Newton recentering direction, however, only a "partial" update of s_n is made. Applying the Sherman-Morrison-Woodbury formula to (3.1) yields

$$s_n = \left(A^T D^2 A\right)^{-1} A^T R + \beta(\epsilon) \left(A^T D^2 A\right)^{-1} c, \qquad (3.2)$$

which is a linear combination of the dual affine direction and a transformed gradient term. (Note that β is a function of ϵ.) The partially updated Newton recentering direction is obtained by only evaluating $A^T R$ and ϵ_n at $\tilde{u}_i$. Thus,

$$\begin{aligned} s_n = &\left(A^T D^2 A\right)^{-1}_{u_i} A^T R_{\tilde{u}_i} \\ &+ \beta(\epsilon_n)_{\tilde{u}_i} \left(A^T D^2 A\right)^{-1}_{u_i} c. \end{aligned} \qquad (3.3)$$

This direction is easily computed using the already factored form of $A^T D^2 A$. Since $A^T D^2 A$ is a positive definite matrix, this updated Newton search direction is a trajectory improving direction.

Our best results for the two-step method were obtained using the updated recentering directions, and it is this implementation that is reported in §4. Our two dimensional subspace methods, discussed next, show even better performance.

3.2. Two-Dimensional Subspace Methods

Observe that the dual affine and recentering directions determine a two-dimensional plane and that this plane intersects the polytope to form a two-dimensional cross section on which the current estimate lies. We obtain a search direction by minimizing the cost function on this cross section. Given two linearly independent directions, s_1 and s_2, the two-dimensional subproblem is thus

$$\begin{array}{ll} \min_{\zeta_1,\zeta_2} & \zeta_1 c^T s_1 + \zeta_2 c^T s_2 \\ \text{subject to} & \zeta_1 A s_1 + \zeta_2 A s_2 \leq b - Au \end{array} \tag{3.4}$$

for scalars ζ_1 and ζ_2. The solution to this subproblem then determines weights for the search directions s_1 and s_2, respectively, that define the multi-directional search direction

$$s = \zeta_1 s_1 + \zeta_2 s_2.$$

The solution to (3.4) produces an optimal search direction with respect to s_1 and s_2 at the current point. Specifying a steplength completes the algorithm.

The only restriction on s_1 and s_2, the generators for the subproblem, is that they be linearly independent. The dual affine direction and the Newton recentering direction produce the obvious choice for the subproblem generators, namely

$$s_1 = \left(A^T D^2 A\right)^{-1} |_{u_i} \quad c$$
$$s_2 = \left(A^T D^2 A\right)^{-1} |_{u_i} \quad A^T R |_{u_i}.$$

The partially updated Newton recentering step, discussed in §3.1, could also have been used to obtain s_2. In our computational studies however, we found that the former produced better results [BDDW88].

We also have examined the properties of a second set of generators,

$$s_1 = \left(A^T D^2 A\right)^{-1} |_{u_i} \quad c$$
$$s_2 = \left(A^T D^2 A\right)^{-1} |_{u_i} \quad a_k,$$

where a_k is the first constraint encountered in the s_1 direction. This choice of generators is motivated as follows. Suppose we have a search direction

$$s_1 = \left(A^T D^2 A\right)^{-1} d_1$$

for some d_1. (In this study, $d_1 = c$ so s_1 is the dual affine direction.) Then let k be the index of the first constraint encountered in the s_1 direction. From the current point, u_i, take a step of, say, 99% of the distance to this constraint, obtaining a point $\tilde{u}_i$. Compute $r_k^{new}(\tilde{u}_i)$. Now a rank-one update of $\left(A^T D^2 A\right)^{-1}$ due to the change in residual k, can be written as

$$(A^T D^2 A)_{new}^{-1} = A^T D^2 A - \frac{a_k a^T_k}{r_k(u_i)^2} + \frac{a_k a^T_k}{r_k^{new}(\tilde{u}_i)^2}.$$

Evaluating the new Hessian inverse using the Sherman-Morrison-Woodbury formula and the previously factored form of $\left(A^T D^2 A\right)^{-1}$ results in a second direction $s_2 = (A^T D^2 A)^{-1}_{new} d$, for any choice of d not orthogonal to a_k. This direction has as a dominant component in the direction $\left(A^T D^2 A\right)^{-1} a_k$, and if $d = d_1$, the new direction is dominated by s_1 and $\left(A^T D^2 A\right)^{-1} a_k$ for the subsequent step.

The generators to the subproblem can be varied depending on the location of the current estimate, i.e., its proximity to the optimal vertex. This is done in order to create a globally effective algorithm and to alleviate problems caused by ill-conditioning of the Hessian.

4. Computational Results

4.1. Methods Analyzed

In this section, we present results for two of the methods described in §3:

- a two-step method comprised of a dual affine step followed by a recentering step; and
- a two-dimensional subspace method.

The results from a dual affine approach are used as the base-line for comparing these more promising methods. It has been shown in [MM87] and [MMS88] that the dual affine method compares favorably to MINOS 5.0 [MS83], a well known and widely available implementation of the simplex method. Since our dual affine implementation reproduces the dual affine results reported in [MM87] and [MMS88], it is assumed that our work would also compare favorably with the MINOS simplex code.

The two-step method is implemented using a dual affine step followed by a steepest descent recentering step in the early iterations, and a dual affine step followed by a partially updated Newton recentering step in the final iterations. The switch from one recentering direction to the other is based on ϵ, the residual to the objective row. The two-step approach is used in both Phase 1 and Phase 2.

The two-dimensional subspace method also uses the residual to the objective row, ϵ, to switch between strategies. While $\epsilon \geq 1$, the two-dimensional subproblem generators are

$$s_1 = s_{da}$$
$$s_2 = \left(A^T D^2 A\right)^{-1} a_k.$$

These generators work well in the early iterations when the number of active constraints is small. Once $\epsilon < 1$, however, we switch s_2 to $\left(A^T D^2 A\right)^{-1} A^T R |_{u_i}$. In both cases, the two-dimensional subproblems are solved exactly using the simplex method (see §4.2).

The two-dimensional subspace methods were originally configured in two ways. The first used the solution to the two-dimensional subproblems in both Phase 1 and Phase 2. The second used a dual affine approach in Phase 1 and did not use the two-dimensional subproblem solution until

Phase 2. The former did not perform as well as the latter and therefore only the results of the latter configuration are reported here. Since the initial feasible solution can have a significant effect on a method's overall performance, we are now investigating Phase 1 procedures other than those described below. (See, e.g., [Bar88].)

4.2. Implementation Details

Starting Values and Initial Feasible Points. For each of the problems analyzed in this study, the initial solution is $u_0 = 0$. A **big-M** Phase 1 procedure (see, e.g., [BJ77]) is used to obtain an initial feasible solution when necessary. This is implemented by adding an artificial variable with coefficient 1 to every row in A. The Phase 1 problem is then solved with an artificial variable with coefficient $M = 10^8$ added to the original objective row. The Phase 2 problem begins once the value of the artificial variable becomes negative and can therefore be removed.

Scaling. In the implementations reported in this paper, the A matrix is not scaled. The two-dimensional subproblem constraint matrix defined by (3.4) has been scaled, however, to improve the numerical stability of the subproblem solution. The two columns of this matrix are constructed using the normalized search directions, $s_1/\|s_1\|_2$ and $s_2/\|s_2\|_2$, respectively (see §3.2). Each row of the subproblem constraint matrix is then scaled to have norm 1.

Constraint Dropping. Constraints that are sufficiently far from the current point u, i.e., those having residuals $r_j(u)$ that satisfy

$$r_j(u) > 10^{12} \times \min\{r_k(u), k = 1, \ldots, m\}, \qquad (4.5)$$

are explicitly removed from the computations. Constraints j that satisfy (4.5) are "dropped" by setting R_j and D_{jj} to zero prior to computing $A^T D^2 A$ and $A^T R$. This improves the sparsity in $A^T D^2 A$ and the numerical accuracy of the resulting search directions, and therefore leads to improved performance.

Steplength Selection. As discussed in §3, the steplength for the dual affine method is generally specified as a large percentage of the distance to the boundary of the polytope. In Table 1, two sets of dual affine results are listed, differing only in the percentage values used. The first set is implemented using the same steplength configuration as that reported in [MM87], i.e., the steplength is 99% of the distance to the boundary of the polytope for the first 10 iterations, and 90% of the distance thereafter. The steplength for the second set of dual affine results is 99% of the distance to the boundary of the polytope for *all* iterations.

The steplengths for the two-step method are specified independently for each of the two search directions. In the the dual affine direction, the steplength is 99% of the distance to the boundary of the polytope. In the recentering direction, the steplength is selected using a standard quadratic line search.

For the two-dimensional subspace procedures, the search direction is determined by the subproblem solution. This solution provides multipliers that take the current estimate to an exterior face of the polytope. The steplength is 99% of the distance to that face.

Solving the Two-Dimensional Subproblem. The two-dimensional subspace methods are solved exactly using the general purpose simplex method implemented in IMSL routine ZX4LP [IMS84] and a dual formulation of the subproblem. Empirically, the number of pivots required for each subproblem was found to be less than 15 in most cases.

Stopping Criteria. Three convergence tests are used to terminate the iterations. Objective function convergence is obtained when

$$\frac{|z_i - z_{i-1}|}{|z_i|} < 10^{-8}$$

where z_i is the objective function value at iteration i. The convergence criterion based on the relative difference between the primal and dual objective values is of the form

$$\frac{|z_i^d - z_i|}{\max\{|z_i^d|, |z_i|\}} < 10^{-8}$$

where z_i^d is the dual objective function value at the current iteration. This, of course, can only be tested when the dual multiplier estimate $D^2 A \left(A^T D^2 A\right)^{-1} c$ is nonnegative (see [MM87]). The third convergence criterion is based on steplength, where convergence is observed when the steplength

$$\Delta_S \leq 10^{-16}.$$

Computing Environment. The methods reported here were implemented in Fortran and executed in double precision on the Cyber 205 at the National Bureau of Standards central computing facility. The A matrix is encoded in sparse format using the XMP experimental mathematical programming data structures described in [Mar81], and the Hessian is encoded and solved using the Yale sparse matrix package SMPAK [SMP85] with non positive definiteness of the Hessian handled in the standard way by augmenting the diagonal entries.

4.3. Test Set Description

The methods analyzed in this study were tested on 31 of the 54 publicly available linear programming problems available on Netlib through [Gay85]. The problems omitted from our study are those with implicit bounds, which our

implementations do not currently handle. All but 3 of the 31 problems analyzed required Phase 1 to obtain an initial feasible point given $u_0 = 0$. Another 8 problems do not have a full dimensional interior and therefore only required Phase 1 to find the optimal solution. The remaining 20 problems required both Phase 1 and Phase 2.

4.4. Observations

Convergence. Our results agree well with the accepted optimal values provided in [Gay85]. With few exceptions, each of our implementations solve the problems in our test set "correctly", converging to the accepted value with at least 7 digits of agreement [BDDW88]. The most noteworthy of the exceptions is problem CzProb. None of the methods reported here, and in fact none of the methods we tested, converge with more than 3 digits of agreement for CzProb, although all of our methods do converge to exactly same value, namely $z_{CzProb} = 2182528.5$.

Excepting CzProb, both variants of the dual affine method agreed with the accepted values for all of the remaining problems. The two-step method, however, failed to agree for one other problem, E226 (relative error = 6e-6), while the two-dimensional subspace method failed to agree with the accepted value for Ship12l (relative error = 3e-5).

We are currently investigating methods for determining the optimal basis from interior point solutions such as these. One option is that of computing the Lagrange multipliers and checking for dual feasibility at suspected optimal solutions. "Restarting" the iterations when the Lagrange multipliers indicate a non-optimal solution has been found should eliminate the problems noted here.

Iteration Counts. Each of the methods reported in this paper have the same order work per iteration. Iteration counts rather than execution times are reported, thus having the advantage of making these results comparable over different machines.

Our results show that the dual affine method using the 99/99 steplength configuration results in an overall reduction in the number of iterations for the problems in our test set when compared to the dual affine implementation using the 99/90 steplength configuration used by [MM87]. While this reduction is generally only an iteration or two, the iteration count for CzProb is decreased by 6 iterations, a relative change of 12%. There are also only two instances where the total number of iterations increased using the 99/99 steplength variant, in one case by 1 iteration and in the other by 2. These results thus indicate that the 99/99 configuration is preferable over the 99/90 configuration. Note that our 99/99 dual affine results also compare favorably with those reported in [MMS88]. We thus use the 99/99 steplength configuration of the dual-affine method as our base-line for comparing the two-step and two dimensional subspace methods.

Our results show that the two-step method results in a decrease in the number of iterations almost 3 times more often than it results in an increase when compared to the dual affine approach with a 99/99 steplength configuration:

- 16 of the problems show a decrease in the number of iterations,
- 6 of the problems show an increase, and
- 9 of the problems show no change.

The maximum relative decrease in the iteration count is 25%, the maximum relative increase is 46%, and, on the average, the relative number of iterations decreases by 2%. There is no obvious difference between the results for the first half and those for the second half of the problem set, indicating that the method performs equally well on both the smaller and larger problems.

The results for the two-dimensional subspace method are significantly better than both the dual affine or two-step methods. Using this method, the number of iterations decreased 10 times more often than it increased:

- 20 of the problems show a decrease in the number of iterations,
- 2 of the problems show an increase, and
- 9 of the problems show no change, of which 8 are Phase 1 problems and therefore cannot show a change. (See §4.1.)

The maximum relative decrease in the iteration count is 41%, the maximum relative increase is 11%, and, on the average, the relative number of iterations decreases by 16% (12% counting the 8 Phase 1 problems in the total number of problems). Again, there is no obvious difference between the results for the first half and those for the second half of the problem set.

4.5. Conclusions

The results of this study demonstrate the computational advantages of using recentering ideas and more sophisticated adaptations to the traditional method of centers. In particular, our two-dimensional subspace procedure produces results that are a significant improvement over the dual affine method, reducing the number of iterations by an average of 16%. The two-dimensional subspace results are also competitive with the dual affine results reported in Monma and Morton [MM87] and with the primal-dual interior point results reported in and McShane *et al.* [MMS88]. The procedures presented do not increase the order of the work required per iteration, and can be implemented easily.

Table 1: Iteration Counts

Name	Dual Affine (99/90) Phase 1/ Total	Dual Affine (99/99) Phase 1/ Total	2-Step Phase 1/ Total	2-D Subspace Phase 1/ Total
Afiro	1 / 21	1 / 20	1 / 20	1 / 13
ADlittle	1 / 22	1 / 21	1 / 21	1 / 21
Scagr7	3 / 24	3 / 23	3 / 23	3 / 21
Sc205	4 / 28	4 / 26	4 / 27	4 / 19
Share2b	4 / 30	4 / 30	5 / 28	4 / 22
Share1b	7 / 40	7 / 37	6 / 36	7 / 32
Scorpion	5 / 25	5 / 23	5 / 25	5 / 19
Scagr25	3 / 27	3 / 27	3 / 27	3 / 28
ScTap1	6 / 36	6 / 34	6 / 33	6 / 28
BrandY	35 / 35	36 / 36	27 / 27	36 / 36
Scsd1	0 / 17	0 / 17	0 / 15	0 / 10
Israel	9 / 40	9 / 39	6 / 57	9 / 38
BandM	7 / 31	7 / 30	7 / 30	7 / 25
Scfxm1	30 / 30	32 / 32	28 / 28	32 / 32
E226	42 / 42	39 / 39	33 / 33	39 / 39
Scrs8	53 / 53	51 / 51	43 / 43	51 / 51
Beaconfd	26 / 26	25 / 25	24 / 24	25 / 25
Scsd6	0 / 19	0 / 19	0 / 19	0 / 14
Ship04s	5 / 29	5 / 28	5 / 27	5 / 25
Scfxm2	37 / 37	36 / 36	36 / 36	36 / 36
Ship04l	4 / 31	4 / 28	4 / 27	4 / 24
Ship08s	5 / 33	5 / 30	5 / 27	5 / 26
ScTap2	6 / 33	6 / 32	6 / 33	6 / 26
Scfxm3	38 / 38	37 / 37	35 / 35	37 / 37
Ship12s	5 / 33	5 / 29	5 / 27	5 / 28
Scsd8	0 / 20	0 / 19	0 / 19	0 / 13
ScTap3	6 / 35	6 / 34	6 / 32	6 / 27
CzProb	3 / 50	3 / 44	3 / 46	3 / 39
25FV47	56 / 56	54 / 54	41 / 41	54 / 54
Ship08l	4 / 28	4 / 27	4 / 27	4 / 25
Ship12l	5 / 30	5 / 27	5 / 31	5 / 30
Total	410 / 999	403 / 954	357 / 924	403 / 863

References

References

ARV86 Ilan Adler, Mauricio G. C. Resende, and Geraldo Veiga. *An Implementation of Karmarkar's Algorithm for Linear Programming.* Manuscript ORC 86-8, Department of Industrial Engineering and Operations Research, University of California, May 1986.

Bar88 Earl R. Barnes. Phase 1 procedures for interior point problems. May 1988. Presented at the ORSA/TIMS meeting in Washington, DC.

BDDW88 Paul T. Boggs, Paul D. Domich, Janet R. Donaldson, and Christoph Witzgall. *Algorithmic Enhancements to the Method of Centers for Linear Programming Programming.* Manuscript, National Bureau of Standards, June 1988.

BJ77 Mokhtar S. Bazaraa and John J. Jarvis. *Linear Programming and Network Flows.* John Wiley and Sons, Inc., New York, 1977.

FM68 Anthony V. Fiacco and Garth P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley and Sons, Inc., New York, 1968.

Gay85 David M. Gay. Electronic mail distribution of linear programming test problems. Mathematical Programming Society COAL Newsletter 13, December 1985.

Gon87 Clovis C. Gonzaga. *Search Directions for Interior Linear Programming Methods.* Preliminary version, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, May 1987.

Hua67 Pierre Huard. Resolution of mathematical programming with nonlinear constraints by the method of centres. In J. Abadie, editor, *Nonlinear Programming*, pages 209 219, North Holland, Amsterdam, 1967.

IMS84 *User's Manual: IMSL Library.* IMSL, NBC Building, 7500 Bellaire Boulevard, Houston, Texas 77036-5085, version 9.2 edition, November 1984. Publication Number IMSL LIB-0009.

Kar84 Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373 395, 1984.

Kar85 Narendra Karmarkar. 1985. ORSA/TIMS Joint National Meeting, Boston MA.

Mar81 Roy E. Marsten. The design of the XMP linear programming library. *ACM Transactions on Mathematical Software*, 7(4):481 497, December 1981.

McC87 Garth P. McCormick. Verification of high order optimality conditions using the polyadic representation of derivatives. *Mathematics of Operations Research*, 12:1 9, 1987.

MM87 Clyde L. Monma and Andrew J. Morton. *Computational Experience with A Dual Affine Variant of Karmarkar's Method for Linear Programming.* Internal Memorandum, Bell Communications Research, March 1987.

MMS88 Kevin A. McShane, Clyde L. Monma, and David Shanno. *An Implementation of a Primal-Dual Interior Point Method for Linear Programming.* Manuscript, School of Operations Research and Industrial Engineering, Cornell University, March 1988.

MS83 Bruce A. Murtaugh and Michael A. Saunders. *MINOS 5.0 User's Guide.* Technical Report SOL 83-20, Stanford Optimization Laboratory, December 1983.

MS86 Nimrod Megiddo and Michael Shub. *Boundary Behavior of Interior Point Algorithms in Linear Programming.* Manuscript RJ 5319 (51679), IBM Almaden Research Center, Tel Aviv University and IBM T. J. Watson Research Center, September 1986.

Ren86 J. Renegar. *A polynomial-time algorithm, based on Newton's method, for linear programming.* Report 07118-86, Mathematical Sciences Research Institute, University of California at Berkeley, June 1986.

SMP85 *SMPAK User's Guide Version 1.0.* 1985.

WBD88 Christoph Witzgall, Paul T. Boggs, and Paul D. Domich. *On center trajectories and their relatives in linear programming.* Technical Report, National Bureau of Standards, April 1988.

AN APPLICATION OF QUASI-NEWTON METHODS TO PARAMETRIC EMPIRICAL BAYES ESTIMATION

David Scott, Université de Montréal

Introduction

This article discusses some numerical methods connected with variance estimation in an empirical Bayes setting. Our principal focus is the iterative EM process of Dempster, Laird, and Rubin (1977) as applied to parametric empirical Bayes problems in which the prior distribution belongs to a regular exponential family of distributions with unknown variance (and possibly other unknown hyperparameters). Because the EM process can converge slowly and because each iteration involves the calculation of a posterior expectation, numerical methods designed to contain the computational burden in this method are of interest. A method which assumes normality of the posterior distribution in order to simplify EM calculations has been proposed by Laird (1978) based on a suggestion by Leonard (1975). The main contribution of our research is the use of a quasi-Newton approximation to the observed information matrix in cases where the Leonard-Laird approximation is used in the EM iterations. We show that in practice the quasi-Newton methods give roughly the same degree of accuracy in variance estimation as Newton-Raphson methods, and can allow considerable savings in computation time in problems where a large number of parameters must be estimated.

Empirical Bayes and the EM process

One of the main contributions of the work on maximum likelihood with missing observations by Dempster, Laird, and Rubin (1977) was their demonstration that the unknown parameters in an empirical Bayes problem could be treated as "missing data" in an overall statistical model and then estimated by applying a general model for ML estimation from incomplete data. Nominally, one estimates the hyperparameters of the overall model; the empirical Bayes parameter estimates fall out as by-products of the hyperparameter estimation. This process, to which Dempster et al. gave the name "EM" to emphasize its iterative use of an expectation of missing values (the E step) to carry out maximum likelihood (the M step) had been used before on many occasions. Dempster et al. showed the generality and usefulness of the EM procedure in situations, like parametric empirical Bayes, in which the connection with the missing data problem had not previously been apparent.

The EM process becomes particularly interesting if the complete statistical model of the parameters and the observations is a member of a regular exponential family, because it then becomes necessary to only calculate the posterior expectation of the sufficient statistic for the hyperparameter, rather than all of the missing data, during the E-step at each iteration. In the corresponding M-step this expected sufficient statistic is used to calculate a new maximum likelihood estimate of the hyperparameter.

The general form of the EM calculations for data sampled from an exponential family is as follows. Let Ψ indicate a vector of hyperparameters, Δ a vector of parameters, and x a vector of observed values. Given Ψ, the density of the parameters, $p(\cdot\,|\,\Psi)$, is assumed to belong to an exponential family. The density of the observations given Δ, $f(x|\Delta)$, will in general depend on Δ but not Ψ. The joint model of x and Δ, ignoring a constant of proportionality, is

$$r(x, \Delta|\Psi) \propto f(x|\Delta)p(\Delta|\Psi) . \qquad (1)$$

If we assume an initial estimate Ψ_0 of Ψ, then the p^{th} EM iteration is:

E-step: Given an estimate Ψ_p, calculate the posterior expectation of the sufficient statistic t for Ψ:

$$t_p = E_\Delta(t|x, \Psi_p) \qquad (2)$$

This calculation will in general also yield an estimate Δ_p of the parameters.

M-step: Given t_p, calculate a new estimate of Ψ using maximum likelihood.

$$\Psi_{p+1} = \sup_{\Psi}(t_p|\Psi) .$$

It generally seems to be the case in EM-type calculations for exponential families that once an E-step has been performed the corresponding M-step is straightforward. The E-step, consisting as it does of a posterior expectation, poses more important numerical problems. The obvious approach is to use numerical integration, which for high-dimensional problems can be very time consuming. Laird (1978) proposed to solve this problem by using an approximation which apparently originated with Leonard (1975) and which has been used in many recent studies using parametric empirical Bayes (e.g., Wong and Mason, 1985; Tomberlin, 1988). We first note that the posterior distribution of Δ given x and Ψ is proportional to (1). Our first assumption is that the posterior mean of Δ is the posterior mode. This mode can be found by optimizing (1) with respect to Δ. Second, we assume that the observed information matrix:

$$-H_p^{-1} = -\left[\frac{\partial^2}{\partial\Delta\Delta^T} \log r(x,\Delta|\Psi)\Big|_{\Delta=\Delta_p}\right]^{-1} \qquad (3)$$

accurately represents the posterior covariance matrix of Δ. If one of the components of Ψ is a variance component for Δ, then it will often happen that the sufficient statistic for this variance component involves the observed information matrix. Laird (1978) notes that the two conditions of (i) symmetry and (ii) posterior covariance matrix equal to $-H_p^{-1}$ are satisfied if we assume that (1) is proportional to a Normal density.

If we use the Newton-Raphson method for

unconstrained optimization as a means for carrying out the calculations in the E-step (2), we have available, at the optimum Δ_p for the p^{th} iteration, the matrix H_p of second derivatives at $\Delta=\Delta_p$. The inverse of this matrix is the negative of the observed information matrix (3).

Our investigation of numerical methods in parametric empirical Bayes concerns the implementation of this approximation. Before proceeding to a discussion of numerical methods, we illustrate the EM calculations using the example of empirical Bayes estimation of the scale parameters in the Bradley-Terry paired comparison model.

An example problem

In this section we present an example of parametric empirical Bayes estimation in a classical statistical paradigm, the method of paired comparisons (Bradley and Terry, 1952). Consider an experiment involving comparisons between a set of K objects by a set of N experimental subjects. Objects i and j, for i,j=1,...,K are compared n_{ij} times; the n_{ij} need not be equal to N and in fact need not be equal to each other. The data consist in a matrix of counts $X=\{x_{ij}\}$, for i,j=1,...,K, in which x_{ij} represents the number of times object i is preferred to object j in the n_{ij} times which these two objects are compared. We assume that there are no ties, hence that $x_{ij}+x_{ji}=n_{ij}$ for all i,j and that $x_{ii}=0$ for all i.

Given n_{ij}, we assume that x_{ij} is distributed according to a Binomial (n_{ij}, π_{ij}) distribution, where π_{ij} is the probability that object i will be preferred to object j in a single comparison. Following Bradley and Terry (1952), we propose the following model for the π_{ij}:

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \theta_i - \theta_j \qquad (4)$$

where the θ_k, k=1,...,K are parameters to be estimated. Since π_{ij} is monotone increasing in $(\theta_i-\theta_j)$, this model unambiguously defines a scale on which we can rank the K objects being compared, i.e., object i is "ranked higher" than object j if and only if $\theta_i > \theta_j$.

The θ_k so defined are only unique up to a change in location, in that a constant can be added to all the θ_k without affecting the value of any of the π_{ij}. A constraint must therefore be placed on the θ_k for them to be estimable. We impose this constraint in the following way: we arbitrarily choose one of the θ_k, which without loss of generality we call θ_K, and then we reparameterize the problem in terms of the K-1 parameters

$$\Delta_k = \theta_k - \theta_K \qquad (5)$$

for k=1,...,K-1. This reparameterization is equivalent to fixing the origin of the scale defined in (1) at an arbitrarily chosen parameter value. Note that we can still rank the K objects using the Δ_k instead of the θ_k, by saying that i is "ranked higher" than j if $\Delta_i > \Delta_j$ for both $i,j\neq K$, and $i\neq K$ is "ranked higher" ("ranked lower") than K if $\Delta_i > 0$ ($\Delta_i < 0$).

Our empirical Bayes approach to estimating the Δ_k is inspired by the approach to estimation in log-linear models given by Laird (1978). We consider that the Δ_i are (iid) Normal $(0,\sigma^2)$, where σ^2 is a variance hyperparameter which must be estimated from the data. Thus the hyperparameter Ψ consists of the single component σ^2. We estimate σ^2 through the EM process, using the Leonard-Laird approximation to the posterior distribution of Δ.

We first establish that the overall statistical model for the "complete data" (x,Δ) belongs to an exponential family. This is not difficult since, as we see presently, the joint density of the observations x given the parameters does not involve the hyperparameter, and the joint density of Δ belongs to an exponential family. Thus the overall statistical model of (x,Δ) must belong to an exponential family.

Under our assumptions, the likelihood of the observations is

$$f(x|\Delta,\sigma^2) = \prod_{i=1}^{K}\prod_{j=i+1}^{K}\binom{n_{ij}}{x_{ij}}\pi_{ij}^{x_{ij}}\pi_{ji}^{x_{ji}} \qquad (6)$$

while the joint density of the parameters is

$$p(\Delta|\sigma^2) = \left(\frac{1}{2n\sigma^2}\right)\exp\left(-\frac{1}{2\sigma^2}\sum_{k=1}^{K-1}\Delta_k^2\right). \qquad (7)$$

The expression for the model of the "complete data" can then be written, after taking logarithms and performing some algebraic manipulations,

$$\log r(x,\Delta|\sigma^2) = -\left(\frac{1}{2\sigma^2}\right)\sum_{k=1}^{K-1}\Delta_k^2 - \frac{K-1}{2}\log\sigma^2 + d(x,\Delta) \qquad (8)$$

where $d(x,\Delta)$ does not depend on the hyperparameter σ^2. Note that the two explicit terms in (8) originate in the prior density (7). Also note that (8) is in a form where the sufficient statistic for σ^2 is readily apparent:

$$s^2 = \sum_{k=1}^{K-1}\Delta_k^2 .$$

The EM calculations for this application become the following. Let σ_0^2 denote an initial estimate of σ^2. The p^{th} iteration of the EM procedure is then:

E-step: Assume that an estimate σ_p^2 of σ^2 is at hand. Calculate

$$s_p^2 = E_{\Delta}(s^2|x,\sigma_p^2) = E_{\Delta}(\Delta^T\Delta|x,\sigma_p^2) \qquad (9)$$

which is the posterior expectation of s^2, the sufficient statistic for σ^2.

M-step: Calculate the maximum likelihood estimate σ_{p+1}^2 of σ^2 given that $s^2=s_p^2$.

This estimate is $\sigma_{p+1}^2 = \dfrac{s_p^2}{K-1}$.

A demonstration that this process must ultimately converge may be found in Dempster et al.

(1977).

The Leonard-Laird approximation allows an important simplification in the E-step calculations (9). Since we assume that $-H_p^{-1}$ is the posterior covariance matrix of Δ, and that the posterior mean of Δ is equal to its posterior mode Δ_p, it is easy to show that

$$E_\Delta(\Delta^T\Delta \mid x, \sigma_p^2) = \Delta_p^T \Delta_p + \text{tr } H_p^{-1} \qquad (10)$$

This expression only involves quantities which are readily available at the termination of a straightforward application of the Newton-Raphson method to find Δ_p.

In the remainder of this article we discuss alternative methods for performing the calculations in (10). Our particular interest centers on what changes occur in the quantity (10) if we use a quasi-Newton approximation to H_p instead of calculating second derivatives. This interest stems from the potential for considerable savings in computation time if this quasi-Newton approximation is applied.

Numerical implementation of Newton-type methods in the EM algorithm

In the application of the EM process to variance estimation in empirical Bayes problems such as the one we describe, the inverse of the matrix of second derivatives of the complete data density (the "observed information matrix") is used in constructing successive estimates of the variance hyperparameter. This matrix is naturally available if one uses the Newton-Raphson procedure to carry out unconstrained optimization when estimating Δ_p. Let $\Delta_p^{(0)}$, $\Delta_p^{(1)}$, $\Delta_p^{(2)}$,... be the approximations to Δ_p which are generated during the Newton-Raphson procedure applied to the computations (10). The n^{th} iteration of this procedure can be written, for n=0,1,2,...,

$$s^{(n)} = -\left[H_p^{(n)}\right]^{-1} g_p^{(n)} \qquad (11a)$$

$$\Delta_p^{(n+1)} = \Delta_p^{(n)} + s^{(n)} \qquad (11b)$$

where $s^{(n)}$ is a search direction, $g_p^{(n)}$ is the vector of first derivatives (with respect to Δ) of $\log r(x,\Delta|\sigma^2)$ evaluated at an estimate $\Delta_p^{(n)}$ of Δ_p, and $H_p^{(n)}$ is the matrix of second derivatives (the Hessian matrix) evaluated at the same point. At convergence of the Newton-Raphson procedure, Δ_p is taken to be the final iterate in the sequence generated by (11) and the observed information matrix H_p^{-1} is the inverse of the Hessian evaluated at Δ_p.

From (10), however, we see that at each E-step in the EM process we do not need the entire observed information matrix but only its trace. We can thus use an elementary result in numerical linear algebra (e.g., Golub and Van Loan, 1983) to avoid having to explicitly calculate the Hessian inverse at all. Any positive definite matrix M can be decomposed as $M=AA^T$ where A is nonsingular and lower triangular. Then $M^{-1}=C^TC$ where $C=A^{-1}$, and $\text{tr } M^{-1} = \|C\|_F$ where $\|.\|$ is the Frobenius norm. This result indicates that the quantity of primary interest for numerical calculation of a variance estimate in the application we are describing is the Cholesky factor of H_p, that is, the lower-triangular matrix L_p such that $L_pL_p^T = H_p$. In addition, since this factor is triangular, its inverse is also triangular and this fact can be incorporated into very fast algorithms for calculating the Frobenius norm of $(L_p)^{-1}$, which is nothing more than the sum of its squared elements.

In fact, modern implementations of the Newton-Raphson procedure do compute the Cholesky factor of the Hessian at each iteration, rather than carry out the calculations exactly as given in (11). A fast, straightforward implementation of the Newton-Raphson procedure can be coded in FORTRAN, for example, using the subroutines for positive definite matrices available in LINPACK (Dongarra, Bunch, Moler, and Stewart, 1979). For our problem, the Cholesky decomposition as coded in LINPACK uses on the order of $(1/6)(K-1)^3$ arithmetic operations, in constrast to the approximately $(K-1)^3$ operations which are necessary to explicitly form the inverse of any of the $H_p^{(n)}$.

A second approach to the calculation of L_p directly approximates L_p using quasi-Newton methods. These methods, also called "variable metric" methods, have a long history of application to the solution of systems of nonlinear equations and unconstrained and constrained optimization. Helpful background on quasi-Newton methods may be found in the review article by Dennis and Moré (1977) and in the text by Dennis and Schnabel (1983).

The basic idea of quasi-Newton methods applied to the nonlinear optimization in each E step is as follows. Given the initial estimate $\tilde{H}_p^{(0)}$ of H_p, we form successive iterates $\tilde{H}_p^{(1)}$, $\tilde{H}_p^{(2)}$,... by

$$\tilde{H}_p^{(n)} = \tilde{H}_p^{(n-1)} + U^{(n)} \qquad (12)$$

where $U^{(n)}$ is a matrix of rank two. $U^{(n)}$ is generally some function of $\Delta_p^{(n)}$, $\Delta_p^{(n-1)}$, $g_p^{(n)}$ and $g_p^{(n-1)}$. Thus a quasi-Newton method does not calculate second derivatives but uses only first-order information. The matrix $\tilde{H}_p^{(n)}$ is then used in place of $H_p^{(n)}$ in the iteration (11) to calculate a new estimate of Δ_p.

A large part of the acceptance of quasi-Newton methods in optimization stems from the ability to directly compute the Cholesky factor $\tilde{L}_p^{(n)}$ of $\tilde{H}_p^{(n)}$ from the Cholesky factor $\tilde{L}_p^{(n-1)}$ of $\tilde{H}_p^{(n-1)}$. Several authors have proposed efficient and numerically stable methods for doing so (see, for example, Gill, Golub, Murray, and Saunders, 1974; Goldfarb, 1976). Importantly, if we have M parameters to estimate these methods can carry out the update in $O(M^2)$ operations, as opposed to the $O(M^3)$ operations required to explicitly form $H_p^{(n)}$ and decompose it. Thus if M is large, quasi-Newton approximations hold a

certain promise for carrying out EM calculations with reduced computational effort.

Two quasi-Newton updates in particular have attracted the attention of researchers. Each has the very useful property that if $\tilde{H}_p^{(n-1)}$ is positive definite, then $\tilde{H}_p^{(n)}$ is as well. These updates are named with the initials of the researchers who initially studied them. The DFP update, named after Davidson, Fletcher, and Powell, is perhaps the best known. The rank-two update (12) characterizing the DFP method is:

$$U_{DFP}^{(n)} = \frac{1}{s^T y}(qy^T + yq^T) - \frac{q^T s}{(s^T y)^2} yy^T$$

where

$$q = g_p^{(n)} - \tilde{H}_p^{(n-1)} s, \quad y = g_p^{(n)} - g_p^{(n-1)}$$

and $s = \Delta_p^{(n)} - \Delta_p^{(n-1)}$. The BFGS update, named for its discoverers Broyden, Fletcher, Goldfarb, and Shanno, is

$$U_{BFGS}^{(n)} = \frac{1}{s^T y}(yy^T) + \frac{1}{s^T p}(pp^T) \qquad (13)$$

where s and y are as defined above and $p = \tilde{H}_p^{(n-1)} s$.

The DFP update was originally devised to yield a good approximation to the analytic Hessian. See Dennis and Moré (1977) for a discussion of the nature of this approximation. The BFGS update is "complementary" to the DFP in the sense that it provides the same sort of approximation to the inverse of the analytic Hessian that the DFP provides to the Hessian itself. Since the calculation (10) involves the negative of a Hessian inverse, we are naturally interested in the BFGS update. In addition, the current consensus seems to be that the BFGS is the most successful quasi-Newton update in practice (see, for example, Dennis and Schnabel, 1983, and the references cited therein).

In this research we have investigated a quasi-Newton optimization method using a BFGS update as an alternative to the Newton-Raphson procedure in the computations leading to (10). In the next two sections we discuss some issues arising from the implementation of this method, and give some numerical results.

Implementation of quasi-Newton techniques

The Newton-Raphson method is well known to practitioners since the method adapts itself readily to a wide variety of problem situations. Quasi-Newton techniques are not as well known and need more careful implementation if they are to be useful. In this section we discuss certain practical problems which arise from the use of quasi-Newton methods, many of which would not be present in an analogous application of Newton-Raphson.

The most important issue is that of finding a good initial approximation to H_p. If the initial estimate of Δ_p is zero, the use of $\tilde{H}_p^{(0)} = I$, the identity matrix, is often a bad choice. In many statistical applications, including the one we have described in this article, there is often an a priori reasonability to using the origin as the initial estimate $\Delta_p^{(0)}$ of the parameter vector, but the likelihood function is often very poorly behaved at the origin and the gradient evaluated there may be large. If this is the case, then by (11) the next iterate $\Delta_p^{(1)}$ may be a very poor estimate of Δ_p. In addition, the quasi-Newton iterations build up approximate second-order information based on calculated first-order information. Thus if the initial estimates of Δ_p are too off the mark then much of the early quasi-Newton updating will be counterproductive.

Determining an initial Hessian approximation is known as "scaling" the optimization problem since experience has shown much better behavior of quasi-Newton techniques if the initial iterate $\Delta_p^{(1)}$ is of roughly the same magnitude as Δ_p. This scaling problem does not come up in the Newton-Raphson method because the iterate $\Delta_p^{(1)}$ is a function of the calculated Hessian at $\Delta_p^{(0)}$. In relatively well-behaved problems, such as those arising in many estimation problems from regular exponential families, this property of Newton-Raphson allows it to create useful iterates at an early stage.

Many texts on practical optimization techniques (for example, Gill, Murray, and Wright, 1981; Dennis and Schnabel, 1983) suggest the use of line searches to mitigate the effect of a naive choice for the initial Hessian approximation. When line searches are used, the second step of the iteration (11) is modified to

$$\Delta_p^{(n+1)} = \Delta_p^{(n)} + \lambda_n s^{(n)} \qquad (11b')$$

where λ_n is a scale value which is determined by first calculating $s^{(n)}$ using (11a) and then carrying out a unidimensional search along $s^{(n)}$ to find a $\Delta_p^{(n+1)}$ which satisfies certain conditions. In principle such searches may be useful. However, most accepted line search procedures involve function evaluations. Since in many statistical applications (such as the one we consider here) the function to be minimized is the log of a product, hence the sum of many logs, the function is extremely expensive to evaluate. The additional expense of function evaluations during line searches may outweigh any efficiency advantages which might be gained from using approximations to the Hessian matrix.

In this research we have used the following procedure for initializing the Hessian approximation:

(a) at the first EM iteration, we take $\tilde{H}_0^{(0)} = H_0^{(0)}$ that is, we initialize the approximate Hessian using the analytic Hessian evaluated at $\Delta = \Delta_0$.

(b) for all subsequent EM iterations we take $\tilde{H}_p^{(0)} = \gamma_p D_p$, where γ_p is a scalar and D_p is a diagonal matrix.

The matrix D_p is constructed following a suggestion by Dennis and Schnabel (1983) that prior knowledge about the magnitude of the parameter estimates should be used to scale the optimization problem. Let $(D_p)_i$ denote the i^{th}

diagonal element of D_p. We set

$$(D_p)_i = \max\left[1, |(\Delta_{p-1})_i|\right]$$

where $(\Delta_{p-1})_i$ is the i^{th} component of Δ_{p-1}. We derive the constant γ_p by considering the form of the iteration (11). By taking first and second derivatives of (8) it can be shown that each of the components of the gradient vector is a sum of K-1 terms, each of which involves a sample size n_{ij}. Furthermore, each of the elements of the diagonal of the analytic Hessian is such a sum. Since we want both sides of (11a) to be roughly on the same scale, we take

$$\gamma_p = \frac{\bar{n}(K-1)}{4} \qquad (14)$$

where

$$\bar{n} = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} n_{ij}$$

is the average sample size. The expression (14) would be equal to each of the diagonal elements of the analytic Hessian evaluated at $\Delta=0$, in the case where $n_{ij}=\bar{n}$ for all i and j.

This procedure for initializing the Hessian approximation is a compromise between using the high-quality, but expensive, scaling information available in the Newton-Raphson procedure and the less expensive, and less reliable, technique of using a diagonal matrix. We do the former when our prior information about Δ is poor, at the first iteration; we do the latter when our prior information about Δ is better.

In fact, we use $\Delta_p^{(0)}=0$ as a starting value only at p=1, the first iteration of the EM. Since the EM process solves a sequence of similar optimization problems, Δ_{p-1} provides a very good estimate of Δ_p. Therefore, for p=2, 3,... we take

$$\Delta_p^{(0)} = \alpha\Delta_{p-1}$$

where $\alpha\in(0,1]$ is a shrinking factor. The shrinking factor is applied to prevent the approximation to Δ_p from being too good, in order to allow sufficient quasi-Newton iterations to build up a reasonable approximation to H_p^{-1} before convergence occurs. A premature convergence of the quasi-Newton iterations may cause the EM iterations not to converge.

We have computed our BFGS updates by applying two independent rank-one updates, according to formula (13), to an LDL^T decomposition of $\tilde{H}_p^{(n)}$ according to algorithm C1 of Gill, Golub, Murray, and Saunders (1974). Other methods are available to perform rank-one updates, and to directly compute a rank-two update in a single subroutine call. Some of these methods may provide better numerical stability in hard problems, but they are slower.

Some numerical results

In this section we present results from a comparison of quasi-Newton and Newton-Raphson methods in carrying out empirical Bayes estimation of the scale parameters of the paired-comparison model. We use both real and simulated data. Using the real data, we show that both methods give virtually identical results. Unfortunately, each of the real data sets is too small to show any computational benefit from using quasi-Newton methods (in fact, the quasi-Newton iterations are much slower). We have therefore simulated large data sets in order to give an idea of the kind of savings in computation time which might be expected from using quasi-Newton techniques on large problems.

In the case of both the Newton-Raphson and quasi-Newton methods we have used initial estimates $\Delta_1^{(0)}=0$ and $\Delta_p^{(0)}=\alpha\Delta_{p-1}$ for p=2,3,... The results reported below use $\alpha=.8$. In testing we used $\alpha=.9$ as well, but interestingly the smaller value of α induced fewer quasi-Newton iterations. The effect of using $\alpha>0$ on Newton-Raphson was to reduce the number of Newton iterations by about a third, although there was no discernible difference in effect between the two values of α.

The Newton-Raphson and quasi-Newton iterations are terminated when

$$\frac{\|\Delta_p^{(n)} - \Delta_p^{(n-1)}\|_\infty}{\|\Delta_p^{(n-1)}\|_\infty} < \varepsilon$$

and the EM iterations are terminated when

$$\frac{|\hat{\sigma}_p^2 - \hat{\sigma}_{p-1}^2|}{\hat{\sigma}_{p-1}^2} < \varepsilon$$

where $\varepsilon>0$ is an error tolerance. We have used $\varepsilon=10^{-6}$.

The simulated data sets were generated by drawing M=K-1 values Δ_k^* from a Normal (0,1) distribution, for M=60, 80, 100, 150, and 200. These simulated parameters were then used in binomial experiments to generate data matrices $\{x_{ij}^*\}$ according to the Bradley-Terry model (4). We have used $n_{ij}=50$ for all i and j in all simulations.

Our numerical results are summarized in Table 1, where we report, for both Newton-Raphson and quasi-Newton methods, the number of EM iterations required, the average number of Newton iterations per EM iteration, the estimated σ^2, and the approximate computation time.

For each of the simulated data sets we compute s_Δ^2, the empirical variance of the generated Δ_k^*. Each of the variance estimates generated by Newton-Raphson and by quasi-Newton approaches in very close to the corresponding s_Δ^2. In all cases, in fact, the Newton-Raphson and quasi-Newton methods give virtually identical variance estimates.

We note that in the case of the simulated data sets, the size of the problem has very little effect on the number of Newton or EM iterations required to converge. In the case of the real and the simulated data, the quasi-Newton method requires 3.5 - 4 times as many iterations per EM iteration. Again, this ratio seems to be independent of the size of the

problem.

The column labelled "QN advantage" gives the ratio of the Newton-Raphson time to the quasi-Newton. For problems in which M, the numbers of parameters, is small, Newton-Raphson is clearly faster. As M gets large, however, the quasi-Newton advantage seems to approach M itself. Such behavior is to be expected, as each Newton-Raphson iteration involves $O(M^3)$ arithmetic operations, while each quasi-Newton iteration involves $O(M^2)$ such operations.

Discussion

In this research we have applied quasi-Newton methods in a parametric empirical Bayes setting where the EM process is used to estimate a variance hyperparameter. We have shown that the variance estimates calculated using our techniques are virtually identical to those calculated using Newton-Raphson methods. In addition, the quasi-Newton methods use substantially less computation time in large problems.

The potential for application of quasi-Newton methods is great in statistics, not only as part of the EM process but more generally. A very fertile area for further research is in the scaling of quasi-Newton optimization when applied to parameter estimation problems. We suspect that the ad hoc scaling solution used in this research will generalize to a rule which may be applied in a wide range of estimation problems.

Acknowledgements

The author is currently on leave from Concordia University. This research has been carried out at the Centre de recherche sur les transports at the Université de Montréal, and has been supported in part by operating grant A6795 from the Natural Sciences and Engineering Research Council of Canada. The author is grateful to K. Brenda MacGibbon for advice and encouragement. Georges Côté and Johanne Gilbert provided computer programming support.

References

Appelby, M.C. "The probability of linearity in hierarchies". Animal Behavior 31, 1983, 600-608.

Bradley, R.A. and Terry, M.E. "The rank analysis of incomplete block designs. I. The method of paired comparisons". Biometrika 39, 1952, 324-345.

Dempster, A.P., Laird, N.M., and Rubin, D.B. "Maximum likelihood from incomplete data via the EM algorithm" (with discussion). J. Royal Statist. Soc., series B, 39, 1977, 1-38.

Dennis, J.E. Jr. and Moré, J.J. "Quasi-Newton methods: motivation and theory". SIAM Review 19, 1977, 46-89.

Dennis, J.E. Jr. and Schnabel, R.B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Englewood Cliffs: Prentice-Hall, 1963.

Dongarra, J.J., Bunch, J.R., Moler, C.B., and Stewart, G.W. LINPACK Users' Guide. Philadelphia: SIAM Publications, 1979.

Gill, P.E., Golub, G.H., Murray, W., and Saunders, M.A. "Method for modifying matrix factorizations". Maths. of Computation 28, 1974, 505-535.

Gill, P.E., Murray, W., and Wright, M.H. Practical Optimization. New York: Academic Press, 1981.

Goldfarb, D. "Factorized variable metric methods for unconstrained optimization". Maths. of Computation 30, 1976, 796-811.

Golub, G.H. and Van Loan, C.F. Matrix Computations. Baltimore: Johns Hopkins University Press, 1983.

Kendall, M.G. Rank Correlation Methods. New York: Hafner, 1962.

Laird, N.M. "Empirical Bayes methods for two-way contingency tables". Biometrika 65, 1978, 581-590.

Leonard, T. "Bayesian estimation methods for two-way contingency tables". J. Royal Statist. Soc. 37, series B, 1975, 23-37.

Tomberlin, T.J. "Predicting accident frequencies for drivers classified by two factors". J. Am. Statist. Assoc. 83, 1988, 309-321.

Wong, G.Y. and Mason, W.M. "The hierarchical logistic regression model for multilevel analysis". J. Am. Statist. Assoc. 80, 1985, 513-524.

Table 1

Numerical Comparison of Quasi-Newton and Newton-Raphson Methods on Selected Data Sets

Data set	No. parameters	s^2_Δ	Newton-Raphson				Quasi-Newton				QN Advantage
			No. EM iterations	Avg. Newton iterations	$\hat{\sigma}^2$	time[3]	No. EM iterations	Avg. QN iterations	$\hat{\sigma}^2$	time[3]	
Dominance[1]	6	N/A	36	4	1.4904	$.1373 \times 10^{-3}$	35	13	1.4321	$.1382 \times 10^{-1}$	0.01
Schoolboys[2]	12	N/A	10	4	1.6781	$.5287 \times 10^{0}$	10	10	1.6781	$.3230 \times 10^{1}$	0.16
(Simulated)	60	0.78	5	4	0.8340	$.5494 \times 10^{12}$	5	14	0.8337	$.2432 \times 10^{12}$	2.3
(Simulated)	80	0.84	5	4	0.8837	$.1795 \times 10^{15}$	5	14	0.8835	$.3019 \times 10^{14}$	5.9
(Simulated)	100	1.20	4	4	1.2235	$.3897 \times 10^{17}$	4	15	1.2232	$.3221 \times 10^{16}$	12.1
(Simulated)	150	0.90	5	4	0.9908	$.2000 \times 10^{21}$	5	15	0.9907	$.1602 \times 10^{19}$	125.0
(Simulated)	200	0.99	4	4	1.0280	$.7676 \times 10^{23}$	4	16	1.0279	$.5317 \times 10^{21}$	144.4

Notes:

1 From Appelby (1983).

2 From Kendall (1962).

3 Times are given in μ sec. All computations were carried out on a SUN 3/50 workstation with floating point acceleration.

NUMERICAL ALGORITHMS FOR EXACT CALCULATIONS OF EARLY STOPPING PROBABILITIES IN ONE-SAMPLE CLINICAL TRIALS WITH CENSORED EXPONENTIAL RESPONSES

Brenda MacGibbon, Concordia & UQAM, Susan Groshen, USC, Jean-Guy Levreault, U. de Mtl

For some cancers, the existing treatment regimens produce long-term disease-free survival rates of 90% or better. In this situation a new protocol may aim to reduce the amount or duration of treatment, while maintaining the high disease-free survival rates. Although the primary goal is to evaluate the specific morbitity of such a new protocol, it is desirable to develop rules to stop the trial if many patients die or relapse early in the study and to study the statistical properties of these rules numerically. Since the failure (death or relapse) or success (survival) of the nth patient is not usually observed before the (n+1)st patient is entered onto the protocol, most developed sequential techniques do not apply to the problem. Most group sequential techniques involve large sample results, inappropriate for small studies. If the survival times of the patients follow an exponential distribution and the entry times into the trial are Poisson, and if these are independent, then a pure birth-and-death process with a well-defined transition matrix is an appropriate model. Analysis of the process enables the expression of error rates in terms of the transition probability matrix and renders these calculations computationally feasible. A conceptually simple design for monitoring a trial, in which a new treatment is evaluated after each observed failure, is presented and algorithms to calculate the error rates of interest are given. Algorithms for the calculation of the average sample number (ASN), the median and the quartiles of the sample size, as a function of the ratio of the entry rate to the failure rate, are constructed. Approximations to these exact results are also given by the use of the ballot problem. Finally, the methods are illustrated on an example involving the design of a pilot study.

1 INTRODUCTION

In the above setting, guidelines or criteria would be useful in helping the investigator to decide when there have been "too many" deaths or failures to justify the continuation of the trial. Two problems may arise in establishing criteria in these situations. The first occurs when the response variable of interest is a time variable such as survival, remission duration, or disease-free survival: the failures (death or relapse) or successes (survival or continuing remission) of the first n patients are not usually observed prior to the entry of the (n+1)st patient into the study. Thus the classical sequential techniques such as those described in Wald [1947], cannot be applied to this situation. The second problem occurs because the expected survival in the proposed study is high or the total number of available patients is limited. Either situation would imply that the observed number of failures will probably be small. Sequential and the more recently developed group sequential techniques appropriate for censored data, or adaptable to censored data, rely on large sample theory for probability calculations (Pocok [1977], [1982], O'Brien and Fleming [1979], Majunder and Sen [1978], Gail [1982], Jennison and Turnbull [1983]). It has been observed (Gross and Clark [1975], Lesser and Cento [1981], Benedetti *et al.* [1982]) that for analyzing censored data, the effective sample size at a given time for a group of n patients, is approximately the number of failures observed prior to that time, and furthermore that asymptotic approximations depend on the number of failures (Selke and Siegmund [1983], Slud [1984], Tsiatis [1982]). Thus asymptotic approximations may not be appropriate under these circumstances.

Any sequential or group sequential procedure appropriate for the one-arm pilot study described above will therefore require exact, finite-sample, probabilities based on a nonparametric method or on a procedure designed for a specific parametric survival distribution. Because of the limited number of failures expected in this setting, nonparametric statistics will be insensitive; parametric techniques, if appropriate, will be more powerful. Since many survival patterns can be well summarized with an exponential curve, several sequential methods have been proposed for the exponential distribution. However none of them are quite suitable for the type of trial under consideration. As demonstrated by Barndorff-Nielsen and Cox [1984], with staggered entry, the distribution of the one-sample likelihood ratio test (and therefore the maximum likelihood ratio estimator) for the parameter of the exponential curve is not explicitly known and is usually approximated using large sample results. Epstein and Sobel [1955] considered one-sample sequential procedures for exponential failure. Their techniques were not appropriate for censoring due to staggered entry and ultimately involved large sample calculations. Breslow and Haug [1972] developed two-sample methods for comparing exponential survival curves which used asymptotic approximations. Canner [1977] employed computer simulations to develop critical regions for a group sequential procedure to compare two survival curves. Klein and Lerche [1983] proposed methods which could lead to exact calculations for the sequential comparison of two exponential survival curves but used large sample approximations to obtain results.

When the survival times are exponentially distributed and entry into the study is Poisson and independent of the survival times, the problem can be modeled as a pure birth-and-death process (see Ross [1980]). This will accommodate censoring due to staggered entry and does not rely on asymptotic theory, thus permitting

one to calculate the exact size and power of any preassigned decision plan. Analysis of the process enables the expression of error rates in terms of the transition probability matrix and renders these calculations computationally feasible. A conceptually simple design for monitoring a trial, similar to one previously proposed by Breslow [1970], in which a new treatment is evaluated after each observed failure, is presented in this paper and the error rates of interest are determined. The average sample number (ASN), the median and the quartiles of the sample size, are calculated as a function of the ratio of the entry rate to the failure rate.

2 THE TESTING PROBLEM AND PROCEDURE

Each patient who will be entered into the trial will be represented by the pair of random variables, (X, Y) where X is the entry time of the patient measured from time zero, the start of the trial, and where Y is the time at which the patient fails, also measured from time zero. We will consider the case where $Y-X$, the survival from entry of the patient into the trial, is exponentially distributed and where the entry into the trial is Poisson. If patient entry is Poisson, then the waiting times are exponential. That is, $X_{i+1} - X_i$ follows an exponential distribution, where X_i is the entry time of the ith patient that comes into the study. Let $1/\lambda_f$ and $1/\lambda_e$ be the expected values of the exponential distributions of the failure times, $(Y_i - X_i)$, and the waiting times between entries, $(X_{i+1} - X_i)$, respectively. The random variables, X_i and $Y_i - X_i$ are assumed to be independent.

If in previous investigations with the intensive or standard therapy, the mean survival time has been $\mu*$, then for ethical reasons, we require that $1/\lambda_f$ based on the modified therapy under consideration be at least as large as $\mu*$. Thus the hypotheses under consideration are:

$$H_O : \lambda_f \leq 1/\mu*$$
$$H_A : \lambda_f > 1/\mu*.$$

The proposed trial design (i.e. the stopping and decision rule) for this testing problem can be summarized as follows: if at any time, the "simple" failure proportion, which is defined to be the observed ratio of number of failures to total number of treated patients exceeds some predetermined threshold (which may depend upon the number of failures) then the trial will be stopped. This is the boundary proposed by Breslow [1970] for binomial responses. More specifically,

1) Plan to enter a maximum of N patients.

2) Establish a threshold, $W*$, such that if the "simple" failure proportion exceeds $W*$, the treatment will be considered ethically unacceptable. This will lead directly to a sequence of critical numbers, $n_1* < n_2* < n_3* < \cdots < n_I* = N$, where n_i* is the smallest integer greater than or equal to $i/W*$. The $W*$ is chosen not only to control error rates, but is also based on ethical considerations which reflect unacceptably high values of λ_f.

3) At the time of each failure, record the number of patients entered on to the tiral. Let n_i be the total number of patients who have begun treatment at the time of the ith failure.

4) If at the ith failure, $n_i < n_i*$, stop the trial and reject H_0. If $n_i \geq n_i*$, continue accruing patients until the next failure is observed or until N patients have been treated.

5) When patient accrual has terminated according to (4) above, then a complete analysis of the data will be undertaken. The rules above are proposed to monitor the study respecting ethical considerations, and not to replace further appropriate analyses.

3 THE BIRTH-AND-DEATH PROCESS MODEL

3.1 Notation and Definitions

Now let us define an event to be either the entry of a patient onto the trial or the failure of a patient. Let the pair (r, j_r) denote the state with exactly j_r failures by the rth event and prior to the $(r+1)$st event. A permissible path will be defined to be a sequence of the pairs $(r, j_r), \{(0,0), (1, j_1), (2, j_2), \cdots (k, j_k)\}$, satisfying $j_1 \leq j_2 \leq, \cdots \leq j_k$ and $r \geq 2j_r$ for all $r = 1, \cdots k$. Let Φ denote the set of permissible paths.

Now define S_i (for $i = 1, 2, \cdots$) to be the subset of all permissible paths that represent trials continued through the $(i-1)$st failure and stopped at the ith failure. Thus $S_i = [p \epsilon \Phi : p = \{(0,0), (1, j_1), (2, j_2), \cdots, (r, j_r), \cdots, (m, j_m)\}$ such that whenever $j_r = i$ it follows that $r \leq n_i* + i - 1$ and whenever $j_r < i$ it follows that $r - j_r \geq n_{j_r}*.]$ A path in S_i must have strictly fewer than j failures at event time $j + n_j* - 1$ (for $1 \leq j < i$) and therefore must have at least n_j* entries into the trial at that time. At event time $i + n_i* - 1$, the path will have at least i failures and no more than $n_i* - 1$ entries into the trial.

The probability of stopping the trial at the ith failure is the sum of the probabilities of all the paths in $S_i, Pr\{S_i\} = \sum_{p \epsilon S_i} Pr\{p\}$. Thus the probability of continuing the trial to the end is $1 - Pr\{\cup_{i=1}^{I} S_i\}$. Let us denote this probability by $P\{C\}$. To calculate $P\{C\}$, we will model this problem as a birth-and death process.

3.2 The Birth-and-Death Process Model

If the assumptions of exponentiality and independence in the preceding section hold, then we have a birth-and-death process where the states are determined by the number of patients alive and on trial (see Ross [1980], Chapter 6, Section 3) and whose transition probability matrix, $P = \{P_{ij}\}$ is given below:

$$(1) P_{01} = 1$$
$$(2) \text{for } 1 \leq i, j \leq N + 1$$

$$P_{i,i+1} = \lambda_e/(i\lambda_f + \lambda_e)$$
$$P_{i,i-1} = i\lambda_f/(i\lambda_f + \lambda_e)$$
$$P_{i,j=0} \text{ for } j \neq i-1 \text{ or } i+1$$
$$P_{N+1,N+1} = 1$$

$P_{i,i+1}$ is the probability that another patient enters the trial prior to any failure when there are i patients on trial and at risk for failure. $P_{i,i-1}$ is the probability that a patient fails prior to another entry, when there are i patients at risk. N is the total number of patients that will enter the trial if early termination does not occur and $I-1$ is the total number of failures permitted prior to the entry of the Nth patient.

P is the transition matrix of a Markov Chain whose states denote total number of patients alive and in the trial, and P_{ij} represents the probability of moving from state i to state j after the occurrence of one event (entry or failure). Let P^r_{ij} be the (i,j)th entry in the product matrix P^r. P^r_{ij} represents the probability of moving from state i to state j after the occurence of r events. Thus $Pr\{(r,j)\} = P^r_{0,r-2j}$.

To calculate the probability of terminating the trial prior to the entry of the initially specified N patients, for given λ_f and λ_e, we will use the transition matrices, P^r, to calculate the exact probabilities of the sets S_i. These transition matrices also enable us to easily compute the average sample numbers (ASN), the usual measure of effectiveness of stopping rules. At the same time, it was felt that the median sample size and the quartile sample sizes could be viewed as a more robust measure of effectiveness of the early stopping mechanism and the transition matrices have also been used to calculate these quantities.

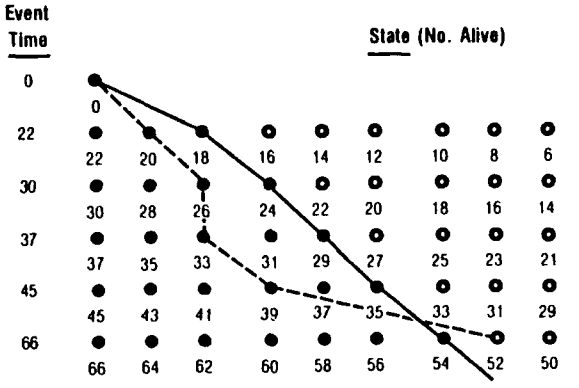
More explicitly, in order to facilitate the discussion of the probability calculations, we will limit ourselves to the following hypothetical trial with the following precise stopping rules (the method can be easily modified for other values of the n_i*'s):

1) Do not plan to stop directly after the 1st or 2nd failure ($n_1* = n_2* = 1$)
2) If the 3rd failure occurs before the 20 th entry, stop; if not, continue ($n_3* = 20$)
3) If the 4th failure occurs before the 27th entry, stop; if not, continue ($n_4* = 27$)
4) If the 5th failure occurs before the 33rd entry, stop; if not, continue ($n_5* = 33$)
5) If the 6th failure occurs before the 40th entry, stop; if not, continue ($n_6* = 40$)
6) If the 7th failure occurs before the 60th entry, stop; if not, continue ($n_7* = 60$)
7) Stop at the 60th entry.

The events, $S_1, S_2, S_3, \cdots S_7$ and C, will be defined as before in Section 3.1. Let $T(i,j)$ denote the outcome of being in the jth state at the ith event time for $i \geq j$.

Figure 1 can be used to visualize permissible paths for the given trial design. Intuitively a permissible path will be a path with a non-positive slope that passes through balls at each of the six stages. An S_3 path will pass from $T(0,0)$ to a white ball at event time 22. An S_7 path will be a permissible path passing through black balls at event times 0, 22, 30 and 37, and a white ball at event time 45.

FIGURE 1 TWO PERMISSIBLE PATHS FOR TRIAL



The mathematical formulation will now be developed. $T(i,j)$, the outcome of being in the jth state at the ith event time ($i \geq j$) actually represents a trial in which there have been exactly $i + (i-j)/2$ entries and $(i-j)/2$ failures. (Note that the $Pr\{T(i,j)\} = 0$ if $i-j$ is not even). Let $\pi[T(i,j) \rightarrow T(k,m)]$ represent the set of permissible paths from $T(i,j)$ $(i \geq j)$ to $T(k,m)$ $(k \geq m)$. The fact that π contains only permissible paths implies that $(i-j)/2 \leq (k-m)/2$. Clearly, if all trials were to be run from the 0th until the 22nd event time, then the $\pi[T(0,0) \rightarrow T(22,2n)]$ for $n = 0, 1, \cdots, 8$, would represent permissible paths which would be stopped by the stopping rule for the 3rd failure. Since $\sum_{n=1}^{11} Pr\{[\pi T(0,0) \rightarrow T(22,2n)]\}$ is equal to 1, and since $\pi[T(0,0) \rightarrow T(22,22)]$, $\pi[T(0,0) \rightarrow T(22,20)]$ and $\pi[T(0,0) \rightarrow T(22,18)]$ represent the sets of all permissible paths of trials not stopped by the stopping rule for the 3rd failure, then the probabilities of stopping or continuing at this stage can be easily calculated using the stochastic matrix, P, defined in Section 3.2.

Let e^i represent the (row) vector, $(0, \cdots, 0, 1, 0, \cdots)$, with a 1 in the $(i+1)$st place and 0's elsewhere. Let $(v)_j$ represent the jth element of the vector contained within the parentheses, v. Now, $Pr\{\pi[T(0,0) \rightarrow T(22,2n)]\}$ can be written as $(e^0 P^{22})_{2n}$, the (2n)th element of the vector $e^0 P^{22}$. More generally, if $\pi[T(i,j) \rightarrow T(k,m)]$ represents the set of permissible paths (that is, $i \geq j, k \geq m, *(i-j)/2 \leq (k-m)/2$, and $(i-j)$ and $(k-m)$ are even) then we have

$$Pr\{\pi[T(i,j) \rightarrow T(k,m)]\} = (e^j P^{k-i})_m \qquad (A1)$$

If $\pi\{T(i,j) \rightarrow T(k,m) \rightarrow T(n,q)]$ is permissible, then its probability is given by

$$(e^j P^{k-i})_m \times (e^m P^{n-k})_q \qquad (A2)$$

The dotted line on Figure 1 joining (0,0), to (22,20)

to (30,26) to (37,33) to (45,39) to (66,52) represents a subset of permissible paths with an endpoint of 59 entries and 7 failures (that is, a trial stopped only at the 7th failure). Its probability is calculated as $(e^0p^{22})_{20} \times (e^{20}P^8)_{26} \times (e^{26}P^7)_{33} \times (e^{33}P^8)_{39} \times (e^{39}P^{21})_{52}$.

Thus the probability of the events $S_1, S_2, \cdots, S_7$, and C, can be calculated exactly, using the stochastic matrix. It suffices to enumerate the permissible paths for each event and to calculate their exact probabilities using extensions of equations (A1) and (A2).

For determining the ASN, the median, and the quartiles of the sample size, the above calculations can be modified to compute, for each n, the probability of stopping the trial after the entry of the nth patient and prior to the entry of the $(n+1)$st patient. To see this, let i be the number of failures such that $n^*_{i-1} < n \leq n^*_i$. The trial will stop after the nth entry only if the ith failure occurs between the nth and $(n+1)$st entries. Therefore the probability of stopping at n patients is the sum of the probabilities of the paths in S_i with exactly $n+i$ events.

3.3 An approximation to the true probability of the events $S_3, \cdots S_I$, C using the ballot problem.

Recall that if we have two independent Poisson processes with parameters λ_1, λ_2 then probability that the nth waiting time in the first process occurs before the mth waiting time in the second process is:

$$\sum_{k=n}^{n+m-1} \binom{n+m-1}{k} \left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right)^k \left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right)^{n+m-1-k}$$

Although our Poisson processes (the entry process and the failure process) are far from being independent, we can use a similar approach if we make the following assumption.

A sequence will be said to be of type $[i, j]$ if it consists of exactly i entries and j failures. In order that a $[i, j]$ type sequence represent a trial, we will require that for each subsequence S_k of length $k \leq i+j$, the number of entries in S_k be greater than the number of failures in S_k. Such a sequence will be called admissible.

Now let us assume that the probability of any sequence of type $[i, j]$ is equally likely (this probability will be denoted $\lambda(i, j)$); standard mathematical techniques can be used to approximate or bound this probability. Under this assumption, the weak sense version of the ballot problem, (*cf* Barton and Mallows (1965)) can be applied and the probability that a sequence of type $[i, j]$ is admissible $= (i+1-j)/(i+1)$ (Let us denote this probability by $q(i, j)$).

Then the probability of n_3 entries before k_3 failures, that is, the probability of not stopping the trial at the first stage is:

$$\sum_{j=n_3}^{n_3+k_3+1} Pr\{E(j, n_3+k_3-1-j)\}$$

where $E(i, j)$ = union of all type $[i, j]$ admissible paths and

Prob $\{E(i, j)\}$ =

$$\frac{\text{Total \#of admissible sequences of type } [i,j]\times\lambda(i,j)}{\text{Total \#of sequences of type } [i+j-k,k]\times\lambda(i+j-k,k)}$$

$$= \frac{\binom{i+j}{i}\, q(i,j)\, \lambda(i,j)}{\sum_{k=0}^{[(i+j)/2]} \binom{i+j}{k}\, q(i+j-k,k)\lambda(i+j-k,k)}$$

where $[(i+j)/2]$= largest integer less than or equal to $(i+j)/2$. Thus, the probability of not stopping the trial at the first state is

$$\sum_{j=n_3}^{n_3+k_3-1} r(j, n_3+k_3-1-j)$$

where $r(j_1, n_3+k_3-1-j)$=

$$\frac{\binom{n_3+k_3-1}{j}\, q(j_1 n_3+k_3-1-j)\ (j_1 n_3+k_3-1-j)}{\sum_{m=0}^{[(n_3+k_3-1)/2]} \binom{n_3+k_3-1}{m} q(n_3+k_3-1-m,m)\lambda(n_3+k_3-1-m,m)}$$

These approximations tend to work reasonably well in practice.

4. AN EXAMPLE OF THE STOPPING RULE FOR GOOD PROGNOSIS PATIENTS WITH OSTEOGENIC SARCOMA

The calculations presented in this manuscript were prompted by the following clinical situation.

One method of treating the bone cancer, osteogenic sarcoma, in a subgroup of children involves intense chemotherapy followed by surgery and then more chemotherapy (see Rosen *et al.* [1982] and Rosen *et al.* [1983]). Examination of the tumor after removal by surgery can identify those patients with tumors which are very sensitive to the pre-operative chemotherapy and have had at least a 90% tumor reduction. These patients with responsive tumors appear to have a reasonalby good prognosis with the probability of disease-free survival estimated to be approximately .85($\pm$ .077= S.E.) at three years from the start of therapy (unpublished update, Rosen [1982]). However, the chemotherapy regimen is quite intense, with both short-term and possible long-term side-effects. A modified treatment protocol was proposed to shorten the duration of the post-operative chemotherapy, in those patients who had experienced at least 90% tumor reduction as a result of the preoperative chemotherapy. The goal was to reduce the severity of the side-effects while maintaining the overall higher probability of disease-free survival.

It was estimated that approximately 12-15 patients a year would be eligible for the study. Since a study lasting over 5 years was not considered practical, it was decided to plan a single-arm study with 60 patients. Although the main objective of the study was to evaluate toxicity and side effects, it was agreed that a mechanism

to monitor the number of disease recurrences as well as deaths was necessary. Monitoring rules, such as the one in Section 2, were proposed. Based on past studies, it was decided that the assumptions of Poisson arrival for treatment, and of exponential failure during the first 3 years after the beginning of therapy, were reasonable and would provide useful approximations to the actual distributions.

The stopping rule proposed in the previous section has its critical values chosen so that the "simple" failure proportions would never exceed 15%. n_1*, and n_2* were set to 1 so that the trial would not stop after one or two early failures. The maximum number of patients was set to 60 and the maximum number of failures permitted was determined by the critical region for a .05 level one-sided test of a binomial parameter, $\pi = .05$ with $n = 60$.

Table 1 presents the probabilities of stopping early for a variety of ratios of entry rate (λ_e) to failure rate (λ_f). For the particular situation under consideration in this example, we would want a high probability of early stopping if the three-year survival were much below 0.80

TABLE 1: PROBABILITY OF STOPPING EARLY*

$\ln(\lambda_e/\lambda_f)$	$(\lambda_e = 10)$	$(\lambda_e = 12)$	$(\lambda_e = 15)$	**
3.20	0.294	0.231	0.160	1.000
3.30	0.331	0.265	0.190	1.000
3.40	0.367	0.301	0.223	1.000
3.50	0.404	0.337	0.257	1.000
3.60	0.441	0.374	0.292	1.000
3.70	0.476	0.411	0.329	1.000
3.80	0.511	0.447	0.365	1.000
3.90	0.545	0.483	0.402	1.000
4.00	0.577	0.517	0.439	1.000
4.10	0.608	0.551	0.474	1.000
4.20	0.638	0.583	0.509	1.000
4.30	0.666	0.614	0.543	1.000
4.40	0.692	0.643	0.576	0.999
4.50	0.717	0.670	0.607	0.997
4.60	0.740	0.696	0.636	0.993
4.70	0.761	0.721	0.664	0.984
4.80	0.781	0.744	0.691	0.969
4.90	0.800	0.765	0.715	0.944
5.00	0.817	0.785	0.738	0.905
5.25	0.854	0.828	0.790	0.744
5.50	0.885	0.863	0.832	0.515
5.75	0.909	0.892	0.867	0.294
6.00	0.928	0.915	0.894	0.139
6.25	0.944	0.933	0.917	0.056
6.50	0.956	0.947	0.935	0.020
6.75	0.965	0.959	0.949	0.007
7.00	0.973	0.968	0.960	0.002

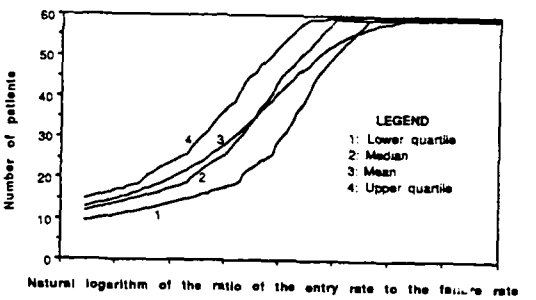
* Columns 2,3,4 correspond to three-year survival corresponding to entry rates of 10, 12 and 15 patients/year respectively.

** Column 5 corresponds to the probability of stopping early.

and we would certainly want a small probability of early stopping if the three-year survival were 0.90 or better.

Figure 2 plots the median, mean, upper and lower quartiles for the number of patients entered for SR3. Once more, the mean and median are similar for the values of $ln(\lambda_e/\lambda_f)$ under consideration. The mean tends to be larger than the median when the probability of early stopping is higher and therefore the distribution is somewhat skewed to the left; the mean tends to be smaller than the median when the probability of early stopping is lower and therefore the distribution is somewhat skewed to the right.

FIGURE 2: Number of patients to be entered into study with stopping rule 1: Expected number, Median, Lower and Upper quartile.



	$\ln(\lambda_e/\lambda_f)$ =	3.5	4.0	4.5	5.0	5.5	6.0	6.5
Probability of 1 year survival	λ_e = 10	0.74	0.83	0.89	0.93	0.96	0.98	0.99
	λ_e = 12	0.70	0.80	0.88	0.92	0.95	0.97	0.98
	λ_e = 15	0.76	0.64	0.85	0.90	0.94	0.96	0.98
Probability of 3 year survival	λ_e = 10	0.40	0.58	0.72	0.82	0.88	0.93	0.96
	λ_e = 12	0.34	0.52	0.67	0.78	0.86	0.91	0.95
	λ_e = 15	0.26	0.44	0.61	0.74	0.83	0.89	0.93

5. CONCLUSION

With the calculations of the entries in the matrices $P^n(n \geq 1)$ for different values of λ_e and λ_f, the exact probability of not stopping early can be computed for a given trial design by summing over the appropriate products as presented in the appendix. With these calculation, the proposed trial design can be evaluated in terms of size, power, ASN, and median sample size. In practice, the patient referral patterns are often known from past experience and thus λ_e may be estimated; different values of λ_f can be used to evaluate the design. This manuscript confined itself to the study of one stopping rule; in MacGibbon *et al.* [1988] several stopping rules are compared and examined according to the above criteria.

Canner [1977] also considered the problem of monitering a trial when the failure was exponential. Using computer simulations for much larger studies, he found that his results were reasonably robust against changing referral patterns, but quite a bit more sensitive to departures from the assumption of exponentially distributed failure data. The effect of varying referral patterns in the setting under consideration is currently being studied.

Since the calculations of the size and power *etc*

are exact if the failure distribution is exponential and if patient entry is Poisson, then for small and moderate sized studies, the proposed sequential stopping rules can be used as exact procedures - - thus establishing the objective criteria to permit the necessary monitoring of one-sample studies.

REFERENCES

Barndorff-Nielsen, O. and Cox, D. (1984) The Effect of Sampling Rules on Likelihood Statistics. International Statistics Review 52:309-326.

Barton, D.E. and Mallows, C.L., (1965) Some aspects of the random sequence. Annals of Mathematical Statistics 36:236-260.

Benedetti, J., Liu, P-Y., Seinfeld, J., and Epton, M. (1982) Effective Sample Size for Tests of Censored Survival Data. Biometrics 69:343-349.

Breslow, N. (1970) Sequential Modification of the UMP Test for Binomial Probabilities. JASA 65:639-648.

Breslow, N. and Haug, C. (1972) Sequential Comparison of Exponential Survival Curves. JASA 67:691-697.

Canner, P. (1977) Monitoring Treatment Differences in Long-Term Clinical Trials. Biometrics 33:603-615.

DeMets, D. (1984) Stopping Guildelines vs. Stopping Rules: A Practioner's Point of View. Communications in Statistics, Series A 13(19):2395-2417.

Epstein, B. and Sobel, M. (1955) Sequential Life Tests in the Exponential Case. Annals of Mathematical Statistics 26:82-93.

Gail, M. (1982) Monitoring and Stopping Clinical Trials, in: Mike, V., and Stanley, K., eds., Statistics in Medical Research: Methods and Issues, with Applications in Cancer Research. J. Wiley and Sons, Inc., NY.

Gross, A. and Clark, V. (1975) Survival Distributions: Reliability Applications in the Biomedical Sciences. J. Wiley and Sons, Inc., NY.

Jennison, C. and Turnbull, B. (1983) Repeated Confidence Intervals for Sequential Clinical Trials. Controlled Clinical Trials 4:33-45.

Klein, H-D. and Lerche, R. (1983) Sequential Tests for Survival Data of Breast Cancer Patients. Mathematical Sciences Research Institute Technical Reports, Berkeley, California.

Lesser, M. and Cento, S. (1981) Tables of Power for the F-test for Comparing Two Exponential Survival Distributions. Journal of Chronic Diseases 34:533-544.

MacGibbon, B., Groshen, S., Cento, S. and Levreault, J.-G. (1988) A Stochastic Model for Exact Calculations of Early Stopping Probabilities in One-Sample Clinical Trials with Censored Exponential Responses (submitted for publication).

Majumdar, H. and Sen, P. (1978) Nonparametric Testing for Simple Regression under Progressive Censoring with Staggering Entry and Random Withdrawal. Communications in Statistics, Series A 7:349-371.

O'Brien, P. and Fleming, T. (1979) A Multiple Testing Procedure for Clinical Trials. Biometrics 35:549-556.

Pocock, S. (1977) Group Sequential Methods in the Design and Analysis of Clinical Trials. Biometrika 64:191-199.

Pocock, S. (1982) Interin Analyses for Randomized Clinical Trials: The Group Sequential Approach. Biometrics 38:153-162.

Rosen, G., Caparros, B., Huyvos, A., et al. (1982) Preoperative Chemotherapy for Osteogenic Sarcoma: Selection of Postoperative Adjuvant Chemotherapy Based on the Response of the Primary Tumor to Preoperative Chemotherapy. Cancer 49:1221-1230.

Rosen, G., Marcove, R., Hyovs, A. et al. (1983) Primary Osteogenic Sarcoma: Eight-year Experience with Adjuvant Therapy. Journal of Cancer Research and Clinical Oncology 106:55-67 (suppl.).

Ross, S. (1980) Introduction to Probability Models, 2nd. Ed., Academic Press, NY.

Sellke, T. and Siegmund, D. (1983) Sequential Analysis of the Proportional Hazards Model. Biometrika 70:316-326.

Slud, E. (1984) Sequential Linear Rank Tests for Two-Sample Censored Survival Data. Annals of Statistics 12:551-571.

Tsiatis, A. (1982) Repeated Significance Testing for a General Class of Statistics Used in Censored Survival Analysis. JASA 77.855-861.

Wald, A. (1947) Sequential Analysis. J. Wiley and Sons, NY.

A Numerical Comparison of EM and Quasi-Newton Type Algorithms for Computing MLE's for a Mixture of Normal Distributions[1]

John W. Davenport
Margaret Anne Pierce
Richard J. Hathaway

Georgia Southern College

ABSTRACT

Calculating maximum-likelihood estimates for a mixture of normal distributions can be one of the most computationally intensive problems in parametric estimation. Maximizing the corresponding likelihood function is complicated by singularities and numerous spurious maximizers. Currently the most popular technique for finding maximizers of the likelihood function is the EM (Expectation Maximization) algorithm. While this iterative algorithm is extremely reliable and usually finds the "good" maximizer from most reasonable initial guesses, it is very slow in cases where the overlap between component normal distributions is great. Another approach, which is faster though thought to be less reliable, is to directly maximize the likelihood function using a (locally) fast iterative algorithm based on some variant of Newton's method. The disadvantage with these quasi-Newton methods is that sometimes the estimate obtained is very dependent on the initial guess used. This paper presents some preliminary numerical results indicating the relative strengths and weaknesses of the EM and quasi-Newton approaches found by testing several methods on a variety of mixture estimation problems. Comparisons made include the computational efficiency and reliability of the approaches tested. The ultimate goal of this research is to learn how the two basic approaches can be hybridized in order to achieve a method that is both quickly convergent and reliable.

1. INTRODUCTION

Normal mixtures are a widely applicable modeling tool whenever the statistical population of interest is itself composed of subpopulations which are distributed according to different normal distributions. A t-variate normal density $p(x|\mu,\Sigma)$, with t-variate mean vector μ and $t \times t$ symmetric, positive definite covariance matrix Σ, is defined for t-variate real x by

$$p(x|\mu,\Sigma)=(\exp(-(x-\mu)^T\Sigma^{-1}(x-\mu)/2)) / ((2\pi)^{p/2}|\Sigma|^{1/2})$$

A mixture of m t-variate normal distributions $p_m(x|\Theta)$ is defined by

$$p_m(x|\Theta) = \alpha_1 p(x|\mu_1,\Sigma_1) + \cdots + \alpha_m p(x|\mu_m,\Sigma_m) \qquad (1)$$

where Θ collectively refers to all the individual component mean and covariance parameters along with the mixing proportions $\alpha_1, \alpha_2, \ldots, \alpha_m$, which must sum to 1 and have values between 0 and 1.

Some of the earlier applications of densities of the form in (1) are from the field of fisheries research, from which we borrow an illustrative example. According to Hosmer (1973), adult halibut of a given age class have lengths distributed according to a mixture of two univariate normal distributions. The lengths of the male halibut are actually normally distributed, as are the lengths of the females, but the two normal distributions modeling the male and female subpopulations are not the same. In this case the overall population of all halibut of a given age class is not normally distributed, but rather is distributed according to a mixture of 2 normals

$$p_2(x|\Theta) = \alpha_M p(x|\mu_M,\sigma_M^2) + \alpha_F p(x|\mu_F,\sigma_F^2) \qquad (2)$$

where for convenience we use M and F to denote parameters corresponding to the male and female subpopulations, respectively. Note that in the notation used above that μ_F, for example, is the mean length of all female halibut of a given age class, while α_M, for example, can be interpreted as the proportion of all halibut of the given age class that are male. Accurate estimates of the complete mixture parameter Θ, based on data $\{x_1,\ldots,x_n\}$ consisting of the lengths of n halibut taken from the population, would be important to scientists interesting in better understanding the population dynamics for halibut. In this paper, we are not interested in the applications, but rather in the computational problem of accurately and efficiently estimating the parameter Θ for mixture densities of the form (1), given a t-variate sample $\{x_1, \ldots, x_n\}$ distributed according to the unknown distribution. An excellent reference for more information on both applications and estimation techniques is Redner and Walker (1984)

The next section contains a brief description of the two types of methods (EM and quasi-Newton) to be tested, and Section 3 contains a description of the numerical tests performed, along with comparisons of the results obtained by the two approaches. The last section contains a discussion of the results along with ideas concerning the successful hybridization of the methods tested.

2. The EM and Quasi-Newton Methods

The theory of maximum-likelihood states that good estimates of the unknown parameter Θ_0 can be obtained from the log-likelihood function $L(\Theta)$, which is defined for a given t-variate sample $\{x_1,\ldots,x_n\}$ and density of the form in (1), by

$$L(\Theta) = \log(p_m(x_1|\Theta)) + \log(p_m(x_2|\Theta)) + \ldots + \log(p_m(x_n|\Theta))$$

The maximum-likelihood theory applied to the normal mixture case asserts that a particular (local) maximizer of $L(\Theta)$ will be a good estimate for Θ_0. This "good" maximizer is denoted here by Θ^*, and the two approaches considered here result from applying different optimization techniques to the problem of finding Θ^* by maximizing $L(\Theta)$.

Both of the optimization algorithms are iterative in nature, and generate, in theory, an infinite *sequence* $\{\Theta_{(r)}\}$ *of approximations to* Θ^*. *The* procedure starts with the user supplying an initial, and usually rough, estimate $\Theta_{(1)}$ of Θ^*. Then the terms of the sequence $\Theta_{(2)}, \Theta_{(3)}, \ldots$, are successively calculated until a particular $\Theta_{(s)}$ is obtained that is close enough to Θ^* to warrant termination of the iteration. Sometimes the iterates never get close to Θ^*, but as a practical matter some stopping criterion must be chosen for each method. The main check used in these tests compares (in a way described in Section 3) the new iterate $\Theta_{(r+1)}$ with the old iterate $\Theta_{(r)}$ at each step. If there is very little difference between the iterates, then this usually means that the accuracy of $\Theta_{(r+1)}$ as an approximation to Θ^* will not be much improved by further iteration. For this reason the implemented iteration is terminated as soon as two successive iterates are very similar.

The differences in the EM and quasi-Newton schemes concern the way that the new iterate $\Theta_{(r+1)}$ is calculated. For the EM algorithm, this is easily described. For $i = 1, \ldots, m$ where m is the number of normal components in the mixture, the following calculations are done for $\alpha_{i(r+1)}$, $\mu_{i(r+1)}$, and $\Sigma_{i(r+1)}$:

$$w_{ik(r)} = \alpha_{i(r)}\, p(x_k|\mu_{i(r)}, \Sigma_{i(r)}) \,/\, p_m(x_k|\Theta_{(r)}) \quad , k = 1,\ldots,n$$

$$A_{i(r)} = \left(\sum_{k=1}^{n} w_{ik(r)} \right)$$

$$\alpha_{i(r+1)} = A_{i(r)} / n$$

$$\mu_{i(r+1)} = \left(\sum_{k=1}^{n} x_k\, w_{ik(r)} \right) / A_{i(r)}$$

$$\Sigma_{i(r+1)} = \left(\sum_{k=1}^{n} (x_k - \mu_{i(r+1)})(x_k - \mu_{i(r+1)})^T w_{ik(r)} \right) / A_{i(r)}$$

The EM algorithm is simpler than methods based on Newton's method, and most of the important theoretical convergence properties of it are given in Redner and Walker (1984). The most important convergence property of EM is that

$$L(\Theta_{(r+1)}) \geq L(\Theta_{(r)})$$

for each iteration, so that progress (in this sense) towards finding a maximizer is always being made. The quasi-Newton methods are general purpose optimization tools, unlike EM, and are much more complicated. An excellent discussion of these methods is in Dennis and Schnabel (1983), and here we only discuss a few of the basic ideas.

Optimization software is generally written to do minimization, but this poses no problem since minimizing $f(\Theta) = -L(\Theta)$ is equivalent to maximizing $L(\Theta)$. Newton's method for generating $\Theta_{(r+1)}$ from $\Theta_{(r)}$ can be interpreted by first building the quadratic model $m(s)$ (of $f(\Theta)$) which is defined by

$$m(s) = f(\Theta_{(r)}) + s^T \nabla f(\Theta_{(r)}) + \left(s^T \nabla^2 f(\Theta_{(r)})\, s \right) / 2.$$

Note that it is a model in the sense that $m(0) = f(\Theta_{(r)})$, $\nabla m(0) = \nabla f(\Theta_{(r)})$, and $\nabla^2 m(0) = \nabla^2 f(\Theta_{(r)})$. Next, assuming that $\nabla^2 f(\Theta_{(r)})$ is positive definite, the model function is globally minimized over s to obtain

$$s_{(r)} = -\nabla^2 f(\Theta_{(r)})\, \nabla f(\Theta_{(r)})$$

which is then used to define the next Newton iterate by

$$\Theta_{(r+1)} = \Theta_{(r)} + s_{(r)} \qquad (3)$$

This basic scheme was varied in two different ways in conducting the tests described in the next section. The variations are clearly noted in reporting the numerical results, but an explanation of them is given here. The first modification is in the definition of the model. Finite difference approximations to the model hessian $\nabla^2 f(\Theta_{(r)})$ and model gradient $\nabla f(\Theta_{(r)})$ are used. This modification makes the general software much easier to use without changing the behavior of the iteration sequence very much. (Other important types of approximations to the Hessian are studied in Dennis and Schnabel (1983).)

The second type of modification involves the incorporation of a global strategy that makes the iteration more reliable. The iteration described in (3) is known to converge rapidly to the solution, when started close enough, but it can many times fail when used some distance away from the solution. Usually, the iteration in (3) is hybridized with a safer iteration so that (3) is used when close to the solution and the safer (and usually slower) iteration is used away from the solution. The three globalization strategies tried in the tests are all described in detail in Dennis and Schnabel (1983) and are called the line search, double dogleg, and hook step. The line search only uses the direction given by the Newton step $s_{(r)}$, while the other two strategies also use the direction given by the negative gradient $-\nabla f(\Theta_{(r)})$. The performance of these different global strategies varies according to the type of problem so that all should be tested when trying to determine the usefulness of a quasi-Newton approach.

We last mention that a reparameterization is used in applying Newton's method to the problem of minimizing $-L(\Theta)$. The constraint $\alpha_1 + \alpha_2 + \cdots + \alpha_m = 1$ is effectively discarded by keeping $\alpha_1, \ldots, \alpha_{m-1}$, and replacing every occurrence of α_m by $1 - (\alpha_1 + \alpha_2 + \cdots + \alpha_{m-1})$.

3. Simulation Runs

Simulation tests were performed using several univariate mixtures of two normal distributions. The choices for the parameters α, μ and σ^2 were as follows:

	Weights (Alpha)			Means (Mu)	
	1	2		1	2
{E}	(.50	.50)	{2}	(0.0	2.0)
{N}	(.20	.80)	{4}	(0.0	4.0)

	Variance (VAR)	
	1	2
{1}	(1.0	1.0)
{T}	(1.0	0.1)

The number or letter in { } is the designation used for the parameter indicated. For example, {E} stands for Alphas of (.5 .5).

All possible combinations of the above parameters were used to generate sample data for the simulation. To facilitate reporting of the results, the following scheme was used to name the data files:

	F	1	E	2	1	0
UNIVARIATE CASE		1				
ALPHA (5 5)			E			
MU (0 2)				2		
VAR (1 1)					1	
TRIAL NUMBER (0-9)						0

Table I: Number of Failures

Good Guess

(There were 10 samples of each type used)

Distribution	EM	Linear Search	Dogleg	Hookstep
1E21	0	[1] 4 (0) [1]	4 [1]	4 [1]
1E2T	10	0 [1]	0 [1]	0 [1]
1E41	0	0	0	0
1E4T	0	(1) [1]	(1) [1]	(1) [1]
1N21	1	7 [1]	7 [1]	7 [1]
1N2T	10	0 [1]	0 [1]	0 [1]
1N41	0	1 (1) [1]	1 (1) [1]	1(1) [1]
1N4T	0	0	0	0

Bad Guess

Distribution	EM	Linear Search	Dogleg	Hookstep
1E21	0	(6) [4]	(4)[5]	(5)[5]
1E2T	10	[10]	[10]	[10]
1E41	0	(4)[6]	(2)[7]	(2)[8]
1E4T	0	(3)[7]	(1)[9]	(2)[8]
1N21	1	(1)[9]	(1)[9]	(1)[9]
1N2T	10	(3)[7]	(1)[9]	[10]
1N41	0	2[5]	1 (3)[6]	(3)[6]
1N4T	0	(1)[7]	(1)[8]	(1)[9]

Note: [1] 4 bad results; (0) very bad results; [1] failure to run

Bad results means that the point of convergence was reasonable but was not "close" to the generating parameters. Very bad results means that the algorithm converged but produced unreasonablely large values. Failure to run means that the algorithm did not converge; therefore, no results were obtained.

Table II: Mean Computational Requirements

Good Guess

Distribution	EM	Linear Search	Dogleg	Hookstep
1E21	[1]63.6	[2]24.9 (33.4)	24.8 (35.1)	24.4 (42.2)
1E2T	42.2	22.2 (32.8)	22.4 (34.6)	23.2 (40.0)
1E41	8.2	19.7 (28.6)	19.1 (29.9)	20.8 (37.2)
1E4T	6.8	13.8 (25.3)	11.7 (25.4)	20.2 (36.0)
1N21	65.2	29.4 (41.6)	29.4 (42.8)	29.6 (51.9)
1N2T	23.3	26.8 (38.7)	26.4 (40.9)	28.3 (46.0)
1N41	10.4	23.3 (33.6)	24.4 (35.6)	24.0 (41.6)
1N4T	5.0	24.7 (35.3)	25.8 (38.1)	24.4 (42.8)

Bad Guess

Distribution	EM	Linear Search	Dogleg	Hookstep
1E21	94.9	40.2 (60.0)	55.0 (66.6)	151.0 (177.8)
1E2T	42.5	-	-	-
1E41	17.1	36.2 (52.7)	52.0 (78.5)	151.0 (171.5)
1E4T	12.5	35.0 (51.5)	36.5 (51.5)	35.5 (45.5)
1N21	99.6	26.0 (35.0)	40.0 (53.0)	151.0 (168.0)
1N2T	32.5	45.0 (74.0)	28.0 (38.0)	-
1N41	21.1	27.8 (50.8)	41.5 (68.2)	89.2 (118.7)
1N4T	10.2	35.0 (66.0)	44.5 (70.0)	151.0 (198.0)

Multiplications and divisions for EM = iterations*14*Sample size(N)

Exponential evaluations in EM = iterations*2*N

Multiplications and divisions in Quasi-Newton = (function calls)*6*N+(gradient calls)*25*N

Exponential evaluations in Quasi-Newton = (function calls + gradient calls)*2*N

Logarithm evaluations = (function calls)*N

[1] 63.6 Iterations

[2] 24.9 Function Calls (33.4) Gradient Calls

Note: Samples were included in the mean only if they ran to completion

IMSL subroutine GGUBFS was used to generate a random number between 0 and 1. This number was compared to ALPHA1, and, if it was smaller than ALPHA1, then distribution #1 was selected to produce the random data item. If the random number was larger than ALPHA2, then distrubution #2 was selected. The data items was then produced by generating a random value on a standard normal using the IMSL routine GGNQF. This standard normal value was translated into an equivalent value from the distribution selected. This process was repeated to produce the 300 data points in each sample. Ten samples were generated for each set of parameters.

All data files were run using Em and Quasi-Newton analysis using two sets of initial values. These initial values were as follows:

GUESS 1: Parameters Used To Create The Input File

GUESS 2:	1	2
ALPHA	.35	.65
MU	$\mu_1 - (\mu_2 - \mu_1)/4$	$\mu_1 + (\mu_2 - \mu_1)/4$
VAR	9	9

4. Conclusions

Test results from Table I show that the EM algorithm converges in every case, even from a poor initial guess, whereas the Quasi-Newton algorithm gives poor to no results when started from a poor initial guess. Table I also indicates that, when started from a good position, the Quasi-Newton algorithm gives better results in the case of unequal variances (see distributions 1E2T and 1N2T). This remained true in the case of an even weight distribution and an uneven weight distribution. However, note that most of the inaccuracy in the EM runs for the case of normals with uneven variances resulted from a poor convergence to the parameter associated with the small variance. As was expected, both algorithms performed well when there was good separation between the normals; however, the Quasi-Newton algorithm did occasionally fail to run. There was no noticable difference in levels of convergence in either of the three Quasi-Newton strategies.

The effectiveness of the EM algorithm is measured by the number of iterations required for convergence (See Table II.). Since the main computational effort incurred in the Quasi-Newton strategies is in computing the function values and/or the gradient values and since each iteration might involve several calls to these processes, we felt that the number of function calls and the number of gradient calls would be a more meaningful statistic to use to measure the effectiveness of the Quasi-Newton strategies. For the samples run, the work required to achieve convergence- in terms of number of multiplications and divisions, number of exponential evaluations and the number of logarithmic evaluations- is slightly more for each of the Quasi-Newton strategies than it is for the EM algorithm (See Table II.). The amount of work for the line search and dogleg strategies was about the same in each case; the hookstep approach required a little more work in each case.

These simulation results suggest that the several directions are worth pursuing. Firstly, the results from these experiments are consist with previous results which indicated that a hybrid strategy is needed which starts with the EM approach and then switchs to a Newton approach or a hill-climbing approach when appropriate. Secondly, a set of experiments similar in nature to these experiments should be tested on multivariate data.

References

Dennis, J.E., Jr. and Schnabel, R.B., Quasi-Newton Methods for Unconstrained Nonlinear Problems, Prentice Hall, Inc., Englewood Cliffs, New Jersey, (1983).

Hosmer, D.W., Jr., "A comparison of iterative maximum-likelihood estimates of the parameters of a mixture of two normal distributions under three different types of samples," Biometrics 29 (1973), 761-770.

Redner, R.A., and Walker, H.F., "Mixture densities, maximum likelihood and the EM algorithm," SIAM Review 26 (1984), 195-239.

This work was partially supported by a Georgia Southern Faculty Research Grant.

HIGHER ORDER FUNCTIONS IN NUMERICAL PROGRAMMING

David S. Gladstein, ICAD Inc.

Introduction

Many mathematical problems are defined as combinations of elementary operations on functions, such as integration, differentiation, and finding roots. Often, however, numerical programs to solve such problems are hand crafted for the particular application, rather than being composed of functionally independent parts [1]. This is largely due to the weakness of traditional algebraic programming languages in manipulating functions.

Languages such as LISP and Scheme [2,3] treat functions as first class objects, and allow one to write *higher order* functions, which have functions as inputs or outputs. This flexibility in the manipulation of functions greatly reduces the distance between the notation of mathematical formulations and the notation of the corresponding numerical program.

Example

A certain problem in sequential analysis deals with the attained significance S of an observed outcome (x, n). $S(x, n)$ is computed as a sum of acceptance probabilities $A(x, i)$, which in turn depend upon a family of density functions $f_i(x)$.

Let ϕ and Φ denote the standard normal density and distribution functions, and let a, b, n, and θ be parameters. Define

$$f_1(x) = \begin{cases} \phi(x-\theta), & (a \le x \le b); \\ 0, & \text{otherwise} \end{cases}$$

$$f_i(x) = \int_a^b f_{i-1}(y)\phi(x-y-\theta)\,dy \qquad (i \ge 2)$$

$$A(x,1) = 1 - \Phi(x-\theta)$$

$$A(x,i) = \int_a^b f_{i-1}(y)\big(1-\Phi(x-y-\theta)\big)\,dy \quad (i \ge 2)$$

$$S(x,n) = \sum_{i=1}^{n-1} A(b,i) + A(x,n)$$

Given x and $0 < l \le u < 1$, it is required to find a confidence interval $[\theta_l, \theta_u]$ with

$$S(x, n|\theta = \theta_l) = l$$
$$S(x, n|\theta = \theta_u) = u$$

Routines for ϕ, Φ, root finding, and integration are required. The first page of listings at the end of this article gives the Common LISP code, namely the functions **normal-density**, **normal-distribution** [4], and **secant** and **romberg** [5]. **secant** finds a zero of a function using the secant method; **romberg** integrates a function of one variable using Romberg's method. These routines were translated into LISP directly from their sources without regard to their eventual application; the code reads like C that happens to be written in LISP.

These routines are sufficient to provide a correct solution, but a major difficulty remains. The f_i are expressed as convolution integrals, each depending upon the previous one until the basis case f_1. Integrating f_i requires evaluating f_{i-1} at many points, at each of which f_{i-2} must be integrated and hence evaluated at many points, and so on. The number of function evaluations would seem to be exponential in i, an unacceptable time complexity. However, it is to be expected that each function f_i might be evaluated repeatedly at certain points. If the function values at these points could be retained, much redundant computation could be avoided.

The higher order function **cacheing** is introduced, mapping functions $f(x)$ to mathematically equivalent functions $g(x)$. g operates by by looking up its input x in a table [6] of $(x, f(x))$ pairs. If the required value is found, it is returned. If not, f is used to compute the value, the table is updated, and the computed value is returned. This method has the desirable property that no prior knowledge of the pattern of repeated evaluations is necessary, making it especially attractive for dynamic programming problems.

Examination of the definition of $A(x, i)$ reveals that it depends upon a, b, and θ, so in some sense it is a function of five variables, not two. On the other hand, in the context in which it is used, namely in the definition of $S(x, n)$, a, b, and θ are fixed and only x and i vary. This conflict is resolved by considering the definition $A(x, i) = \ldots$ to be the definition of a higher order function $\tilde{A}(a, b, \theta)$ which maps values of a, b, and θ to some particular function $A(x, i)$. A subsequent reference to $A(x, i)$ is then understood to indicate such a function A, rather than the functional $\tilde{A}$.

The LISP code for the application appears on the last page of this article. The function

acceptance-probability-function corresponds to $\widetilde{A}$, mapping a, b, and θ to a function of x and i. In addition, it takes n as an input, to determine how many of the functions f_i need be constructed.

Before $A(x,i)$ can be constructed, $\widetilde{A}$ must first build up the array of functions f_i. Each f_i after the first involves the construction of three functions at run time: first the integrand $f_{i-1}(y)\phi(x-y-\theta)$ is created programmatically, it is integrated from a to b with respect to y, leaving x as a parameter, and finally the integral is mapped onto a cacheing version to prevent redundant integration. With the f_i in place, $A(x,i)$ is simply constructed as the integral of a programmatically created function which involves f_i.

The function $S(x,n)$ is simple enough that the LISP function **significance** is barely higher order, taking a function $A(x,i)$, and x, n, and b as inputs. While the argument for treating the definition $S(x,n) = \ldots$ as the definition of a higher order function $\widetilde{S}(A(x,i),b)$ could be applied, there is little to be gained by doing so.

The function **confidence-interval** takes values for x, n, a, b, l, and u, and finds θ_l and θ_u so that $S(x,n|\theta=\theta_l) = l$ and $S(x,n|\theta=\theta_u) = u$. The similarity of the two subproblems leads to the introduction of the internal function **get-theta**, which finds the θ corresponding to a particular value of p. Since the secant method only finds zeros of functions, **get-theta** constructs the appropriate function programmatically.

Discussion

A solution to a non-trivial numerical computing problem has been developed, where the constituents of the LISP code of the solution were transliterations of either standard numerical techniques or statements of the problem. The original solution to this problem was a C language program of about five hundred lines, which took approximately one week to write and debug. The LISP version was written in one half day, and consists (using normal indenting) of less than one hundred lines of code, more than half of which is reusable.

The major cost of software is the time to write and debug it. The more compatible the language of the problem statement and the language of the implementation, the less work in translating from one to the other, and the less opportunity for errors and confusion. Since LISP caters to the creation and manipulation of functions, and mathematical problems are often posed in terms of the definition and use of functions, the use of LISP for numerical work is natural.

Acknowledgements

I would like to thank David Place, Martin Plotkin and Marty Wagner for many helpful comments.

Literature Cited

[1] Halfant, M. and G. J. Sussman. *Abstraction in numerical methods.* Proceedings of the 1988 ACM conference on Lisp and functional programming. ACM Press, New York, NY. 1988.

[2] Steele, G. L. *Common LISP: The language.* Digital Press, Billerica, MA. 1984.

[3] Abelson, H. and G. J. Sussman, with J. Sussman. *Structure and interpretation of computer programs.* MIT Press, Cambridge MA. 1985.

[4] Abramowitz, M. and I. E. Stegun. *Handbook of mathematical functions.* National Bureau of Standards, Washington DC. 1972.

[5] Burden, R. L. and J. D. Faires. *Numerical analysis.* Prindle, Weber & Schmidt, Boston, MA. 1985.

[6] Knuth, D. E. *The art of computer programming, volume 3: Sorting and searching.* Addison-Wesley, Reading MA. 1973.

```
(defvar 1/rad2pi (/ (sqrt (* 2 pi))))

(defun normal-density (x)
  (* (exp (* x x -0.5d0)) 1/rad2pi))

(defvar normal-coefficients
        '(0.0498673470d0 0.0211410061d0 0.0032776263d0 0.0000380036d0 0.0000488906d0 0.0000053830d0))

(defun normal-distribution (x)
  (if (< x 0)
      (- 1 (normal-distribution (- x)))
    (- 1 (* 0.5
            (expt (1+ (let ((xpow x)
                            (sum 0))
                        (dolist (coeff normal-coefficients      ; do for each coeff in normal-coefficients,
                                       sum)                     ; when done return sum
                          (setq sum (+ sum (* xpow coeff)))
                          (setq xpow (* xpow x)))))
                  -16)))))

(defun secant (f p0 p1 &optional (epsilon 1d-8) (imax 20))
  (let ((q0 (funcall f p0))
        (q1 (funcall f p1))
        p)
    (dotimes (i imax                                        ; 0 <= i < imax
               (error "Iteration count exceeded."))         ; error if loop runs out
      (setq p (- p1 (* q1 (/ (- p1 p0) (- q1 q0)))))
      (when (<= (abs (- p p1)) epsilon)
        (return p))                                         ; return root
      (setq p0 p1)
      (setq p1 p)
      (setq q0 q1)
      (setq q1 (funcall f p)))))

(defvar romberg-size 35)
(defvar expt4 (let ((arr (make-array romberg-size)))
                (dotimes (i romberg-size        ; 0 <= i < romberg-size,
                           arr)                 ; when done, return arr
                  (setf (aref arr i) (float (expt 4 (1- i)) 1.0d0)))))

(defun romberg (f a b &optional (epsilon 1d-8))
  (let ((r1 (make-array romberg-size))
        (r2 (make-array romberg-size))
        (h (- b a)))
    (setf (aref r2 1)
          (* h (+ (funcall f a) (funcall f b)) .5))
    (do ((i 2 (1+ i)))                                  ; 2 <= i,
        (nil)                                           ; do forever (until a return)
      (let ((temp r1))
        (setf r1 r2)
        (setf r2 temp))
      (setf (aref r2 1)
            (* .5 (+ (aref r1 1)
                     (* h (do* ((kmax (expt 2 (- i 2))) ; upper bound
                                (k 1 (1+ k))            ; 1 <= k
                                (sum 0))                ; sum accumulates
                               ((> k kmax) sum)         ; k <= kmax, when done return sum
                             (setq sum (+ sum (funcall f (+ a (* (- k .5) h))))))))))
      (do ((j 2 (1+ j)))                                ; 2 <= j
          ((> j i))                                     ; j <= i
        (setf (aref r2 j)
              (/ (- (* (aref expt4 j) (aref r2 (1- j)))
                    (aref r1 (1- j)))
                 (1- (aref expt4 j)))))
      (setf h (/ h 2))
      (when (and (>= i 3)
                 (<= (abs (- (aref r2 (1- i)) (aref r2 i))) epsilon)
                 (<= (abs (- (aref r1 (1- i)) (aref r1 (- i 2)))) epsilon))
        (return (aref r2 i))))))                        ; return integral

(defun cacheing (f)
  (let ((cache (make-hash-table :test #'equal)))
    (function
      (lambda (x)
        (or (gethash x cache)
            (setf (gethash x cache) (funcall f x)))))))
```

```
(defun acceptance-probability-function (n a b theta)
  (let ((densities (make-array (1+ n))))
    (setf (aref densities 1)
          (function
            (lambda (x)
              (if (<= a x b)
                  (normal-density (- x theta))
                0))))
    (do ((i 2 (1+ i)))                          ; 2 <= i
        ((>= i n))                              ; i < n
      (let ((previous-density (aref densities (1- i))))
        (setf (aref densities i)
              (cacheing
                (function
                  (lambda (x)
                    (romberg
                      (function
                        (lambda (y)
                          (* (funcall previous-density y)
                             (normal-density (- x y theta)))))
                      a b)))))))
    (function
      (lambda (x i)
        (if (= i 1)
            (- 1 (normal-distribution (- x theta)))
          (romberg
            (function
              (lambda (y)
                (* (funcall (aref densities (1- i)) y)
                   (- 1 (normal-distribution (- x y theta))))))
            a b))))))

(defun significance (acceptance-function x n b)
  (+ (do ((i 1 (1+ i)))                         ; 1 <= i
         (sum 0))                               ; sum accumulates
        ((>= i n) sum)                          ; i < n, when done return sum
      (setq sum (+ sum (funcall acceptance-function b i))))
     (funcall acceptance-function x n)))

(defun confidence-interval (&key x n a b
                            (lower 0.05) (upper 0.95)
                            tl0 tl1 tu0 tu1)
  (labels ((get-theta (p-value guess0 guess1)
             (secant
               (function
                 (lambda (theta)
                   (- (significance
                        (acceptance-probability-function
                          n a b theta)
                        x n b)
                      p-value)))
               guess0 guess1)))
  (let* ((mean (/ x n))
         (rad_mt (sqrt n))
         (tl0 (or tl0 (- mean (/ 1.75 rad_mt))))
         (tl1 (or tl1 (- mean (/ 2.75 rad_mt))))
         (tu0 (or tu0 (+ mean (/ 1.00 rad_mt))))
         (tu1 (or tu1 (+ mean (/ 1.75 rad_mt)))))
    (values (get-theta lower tl0 tl1)
            (get-theta upper tu0 tu1)))))
```

THEORY OF QUADRATURE IN APPLIED PROBABILITY: A Fast Algorithmic Approach

Allen Don, Long Island University

ABSTRACT

The integral representation of the moments of a useful class of probability density functions is cast in a canonical form in terms of Gauss-Laguerre quadrature. This transforms the continuous integration into a sum of discrete terms, effectively removing the integral sign and exposing the parameters to numerical investigations. This allows moments from data to be related to the unknown parameters via a system of non-linear equations. This system is easily and quickly solved for the unknown parameters by any of the numerous non-linear equation algorithms available for personal computers and main-frames. In addition, the factorials and gamma functions found in closed form theoretical moment expressions and in density functions are discretized in the same manner, enabling unknown parameters within the arguments of the gamma to be included in numerical searches. A dominant ratios method is introduced for determining initial conditions for the system of non-linear equations to overcome the notable lack of convergence found in non-linear system algorithms when initial conditions are not well-chosen. The theory is connected to reliability problems to show a fast algorithmic approach rather than the usual graphical approach to parameter identification of density functions both for truncated and for full data.

1.0 INTRODUCTION

1.1 The Problem

Determination of parameters for probability distributions from data is hindered by the analytical expressions for the moments being under the integral sign,

$$(1.1)\qquad M_n=\int_0^{\infty} x^n f(x)dx$$

for full moments, and

$$(1.2)\qquad M_{ns}=\int_0^{T} x^n f(x)dx$$

for truncated moments. Otherwise, it might be possible to have a fitting scheme by a system of equations representing the moments on one side of the system, and the data representing the moments on the other side. In addition, the density function f(x), in a number of useful distributions involves the gamma function $\Gamma(n,m,b)$ or factorial in which the parameters and moment designation are arguments within the gamma sign, hence are intractable for use as variables. Thus, the inverse problem, that of obtaining the parameters of a distribution, given the moments calculated from data, is rendered difficult. This is typified by the Weibull distribution which has both problems. As in all continuous distributions, the moment expression is under the integral sign; in addition, the closed form solution for full moments contains the gamma function with the shape parameter as an argument within the gamma sign. For the Weibull, the n^{th} moment about the origin is

$$(1.3)\quad M_n=\int_0^{\infty} abx^{b-1}e^{-ax^b}x^n dx=\left(\frac{1}{a}\right)^{\frac{n}{b}}\Gamma\left(1+\frac{n}{b}\right)$$

The gamma density function has a slightly different problem: the parameters are both under the integral and within the gamma function argument. Note that $\Gamma(m+1) = m!$ when the argument is integral.

$$(1.4)\quad M_n=\int_0^{\infty}\frac{a^{m+1}x^m e^{-ax}x^n dx}{(m+1)}=\frac{(m+1)(m+2)\ldots(m+n)}{a^n}$$

where order=m+1, and M_n represents the n^{th} moment about the origin. Clearly, the full moment expression has a closed form solution which is easily solved by a system of nonlinear equations:

$$(1.5)\quad M_n=\frac{(m+1)(m+2)\ldots(m+n)}{a^m}$$

The sublety in using this expression is in the method of choosing the "extra" value of a in setting up the system of non-linear equations. As an example, with three moments, two values of a and one value of m can be obtained, but a third value of a is required in the non-linear formulation. Thus, the three equations are:

$$(1.6)\quad M_1=\frac{m+1}{a_1}$$

$$(1.7)\quad M_2=\frac{(m+1)(m+2)}{a_2^2}$$

$$(1.8)\quad M_3=\frac{(m+1)(m+2)(m+3)}{a_{gm}^3},$$

where the geometric mean of a_1 and a_2 is

$$(1.9)\quad a_{gm}=(a_1a_2)^{1/2}.$$

While the closed form solution to the gamma's full moments is simple enough as seen above, it will be seen later that the truncated moments have the "order" parameter within the argument of gamma function, and therefore susceptible to the same approach as the Weibull.

For pedagogic reasons, the gamma distribution and Erlang distribution will appear herein to be identical except that the order parameter of the Erlang will be understood to be integer whereas the equivalent for the gamma is understood to be real. Occasionally, the Erlang nomenclature might be used when, in fact, the search for the order parameter yields a real number for the best fit.

A common approach to determination of parameters of distributions in reliability is to use judgement in selecting the model to which data is to be fitted, then use probability paper for that model. A straight line on the

probablity paper indicates that the correct choice has been made. Computer power, with an engineering work-station becoming commonplace at the engineers finger tips, makes it is appropriate to implement probability research of all kinds in a more automated manner.

2.0 BASIC THEORY

2.1 General Principles

Quadrature effectively removes the integral sign and exposes the moment expression and its parameters to numerical search methods,

$$(2.1)\quad M_n = \int_0^{\infty} x^n f(x,m,a,b)dx = \sum_{i=1}^{p} C_i f(x_i,m,a,b)x_i^n$$

where C_i and x_i are Christoffel weights and knots for Gauss-Laguerre quadrature [1, p.105], and p is the number of points(weights and knots).

The arguments under the gamma symbol are "uncovered" in a like manner since the gamma and factorial functions are, in fact, derived from integral expressions as in (2.4) below.

A system of non-linear equations is set up,

$$(2.2)\quad \sum_{i=1}^{p} C_i f(x_i,m,a,b)x_i^n = M_n, \quad n=1,2,\ldots k$$

where k is the number of equations, hence also the number of unknowns being sought. Therefore, k data moments must be used.

The right hand side of the system is the data(moments calculated from data). The left hand side of the system is the quadrature representation of the integral moment expressions.

Moments about the origin are used. M_o is the zeroth moment representing the cumulative distribution function. Hence, for truncated data applied to reliability theory, M_o represents the fraction of units failed.

Gauss-Laguerre quadrature transforms the continuous integration of functions of the kernel e^{-t} to a summation of discrete values,

$$(2.3)\quad \int_0^{\infty} e^{-t}t^n dt = \sum_{i=1}^{p} C_i t_i^n ,$$

which is exact for n=2p-1, when n is integer.

Therefore, the gamma function is related to Gauss-Laguerre quadrature by

$$(2.4)\quad n = \int_0^{\infty} e^{-t}t^{n-1}dt = \sum_{i=1}^{p} C_i t_i^{n-1} = (n-1)!$$

where n can be real or integer.

If n were unknown and had the numerical values of the right-hand-side of the above been given, then, using the Gauss-Laguerre table for the weights (Christoffel numbers) and knots (points on the time axis) and using a system of non-linear equations solved by a suitable algorithm, the value of n, the argument within the gamma sign, would be obtained.

A useful class of probability density functions, the gamma, Erlang, Weibull, Rayleigh, and chi-square, can be cast in a canonical form in terms of Gauss-Laguerre quadrature weights (Christoffel numbers) and knots. In addition, the limitation of the usual (0,∞) interval is removed providing an opportunity to apply this method to other limits of integration, hence, to truncated moments and to probability table generation. Further, the relationship between the normal and chi-square is exploited to generate real-time probability tables for the normal and for sums (convolution) of normals.

2.2 The Stepping Up Concept

While the familiar parameter of the Erlang form of the gamma function is integer, searches will pass through and most likely result in a fractional or real number argument. Also, the unknown gamma arguments being sought are already fractional for other density functions. Accuracy impairment resulting from non-integral arguments and fractional arguments is remedied by increasing the argument in unit steps while simultaneously externally multiplying by a compensating factor related to the step increases as in (2.5) below. The gamma function identities for integers and for stepping-up the argument are well known and found in the most abbreviated mathematical tables. The concept of using these identities or similar identities either in integral or in non-integral arguments as a method of increasing accuracy when used with quadrature methods, is not well known, if at all. The well-known identities are:

$$(2.5)\quad \Gamma(n) = \frac{\Gamma(n+1)}{n} = \frac{\Gamma(n+2)}{n(n+1)} = \frac{\Gamma(n+3)}{n(n+1)(n+2)} = \ldots\ldots\ldots$$

The derivation of the gamma function identities for integers can be obtained from repeated integration-by-parts [2,pp.201-203]. The derivation for non-integral gamma function identities is achieved by the same method. As can be seen from the following integration-by-parts in (2.6), there is no real or integer restriction on n . This observation is necessary as a precursor to an important sublety: ill-conditioning occurs when fractional powers are encountered resulting in the algorithm wandering without ever converging to a solution. Hence, the gamma relationships of (2.5) must be used to place the powers to which the knots are raised within a range in which the quadrature method will work; this is demonstrated by (2.10).

The usual integration-by-parts derivations seen in texts is in a stepping-down mode,

$$(2.6)\quad \int_0^{\infty} t^n e^{-t}dt = -t^n e^{-t}\Big|_0^{\infty} + n\int_0^{\infty} t^{n-1}e^{-t}dt.$$

While the first term on the right above,

$$-t^n e^{-t}\Big|_0^{\infty}$$

vanishes on an interval of (0,∞), it becomes the important term later in truncated distributions and truncated moments. (2.5) stepping-up is a rearrangement of (2.6) stepping-down.

The important subtlety with respect to fractional powers of n can be demonstrated by examining the integral and related quadrature equivalent for n=1/2 = 0.5 .

$$(2.7)\quad (0.5) = \int_0^{\infty} e^{-t}t^{n-1}dt \Big|_{n=0.5} = \sum_{i=1}^{p} C_i t_i^{-0.5}$$

Observe that, without a step-up, the power of t_i is negative. Even with a step-up of one, we find,

$$(2.8)\quad \Gamma(0.5) = \frac{\Gamma(1.0+0.5)}{0.5} = 2\sum_{i=1}^{p} C_i t_i^{0.5}$$

which yields a fractional power of t_i, albeit positive. Christoffel numbers and knots are derived from a positive integer formulation,

$$(2.9)\quad \int_0^{\infty} t^n e^{-t}dt = \sum_{i=1}^{p} C_i t_i^{n}\ ,\ n=\text{integer:}0,1,2..$$

Hence, quadrature is not valid for n<0 even though the gamma function itself is valid. Thus, for a fractional gamma argument, the stepping-up procedure must be initiated from the beginning simply to bring the argument within the range of validity for Gauss-Laguerre quadrature, to wit;

$$(2.10)\quad \Gamma(0.5) = \frac{\Gamma(1.0+0.5)}{0.5} = \frac{\Gamma(2.5)}{(0.5)(1.5)}$$
$$= \frac{\Gamma(3.5)}{(0.5)(1.5)(2.5)}$$
$$= \frac{1}{(0.5)(1.5)(2.5)} \sum_{i=1}^{p} C_i t_i^{2.5}$$

While it appears that the term $t^{0.5}$ of (2.8), has been brought within a range of validity, the errors are quite large. A combination of additional step-ups and an increased number of points (weights and knots) must be used. If the density function contains an m^{th} order polynomial, and the number of step-ups equals s and we are dealing with the n^{th} moment,

$$(2.11)\quad M_n = \int e^{-t} f(t^m) t^n dt\ ,$$

then $m+n+s \le 2p-1$; hence,

$$(2.12)\quad p \ge \frac{m+n+s+1}{2}$$

Therefore, p is the least number of points that must be used. However, for improved accuracy, it is always wise, and simply accomplished, to go beyond this minimum number of points.

2.3 Application to Density Functions

Introduction of the parameter a into the kernel e^{-at} and using a change of variable x=at, the integration provides the following quadrature expression,

$$(2.13)\quad \int_0^{\infty} e^{-at}t^n dt = \int_0^{\infty} e^{-x}\left(\frac{x}{a}\right)^n \frac{dx}{a}$$
$$= \sum_{i=1}^{p} \frac{C_i}{a}\left(\frac{x_i}{a}\right)^n$$

so that the weights and knots can be used directly from the Gauss-Laguerre tables, divided by the parameter a .

Further, in the entire family of exponentially related density functions, the parameter a by which C_i is divided, i.e., C_i/a, is cancelled; hence, the weights C_i are precisely those from the table. This can be seen by using (2.13) together with the gamma formulation of (1.4)

$$(2.14)\quad M_n = \int_0^{\infty} \frac{a^{m+1}x^m e^{-ax}x^n dx}{\Gamma(m+1)}$$

$$= \frac{a^{m+1}}{m!} \sum_{i=1}^{p} \frac{C_i}{a}\left(\frac{x_i}{a}\right)^{m+n}$$

$$= \frac{a^m}{m!} \sum_{i=1}^{p} C_i\left(\frac{x_i}{a}\right)^{m+n} = \frac{1}{m!} \sum_{i=1}^{p} C_i t^m \left(\frac{x_i}{a}\right)^n$$

2.4 Change of Variable leading to Canonical Form

The Weibull distribution becomes tractable by two changes of variables: $u=t^b$ and x=au,

$$(2.17)\quad f(t) = abt^{b-1}e^{-at^b} = ae^{-au} = ae^{-x}.$$

$$(2.18)\quad M_n = \int_0^{\infty} t^n abt^{b-1}e^{-at^b}dt$$

$$= \int_0^{\infty} u^{n/b} ae^{-au}du = \int_0^{\infty} \left(\frac{x}{a}\right)^{n/b} e^{-x}dx$$

For convenience in manipulation and programming, it is useful to use the reciprocal of b,

i.e., $r = \frac{1}{b}$, so that

$$(2.19)\quad M_n = \int_0^{\infty} \left(\frac{x}{a}\right)^{n/b} e^{-x}dx = \sum_{i=1}^{p} C_i \left(\frac{x_i}{a}\right)^{nr}$$

Clearly, when r=1, i.e., b=1, this is the exponential distribution (Erlang order=1). So the relationships are sufficiently similar to give an indication that a canonical form is possible.

The Erlang (gamma) distribution requires only the simple change of variable x=at to change from the integral form to the quadrature form, so that

$$(2.20)\quad M_n = \int_0^{\infty} \frac{a^{m+1}x^m}{\Gamma(m+1)} e^{-ax}x^n dx$$
$$= \frac{1}{m!} \sum_{i=1}^{p} C_i t_i^m \left(\frac{t_i}{a}\right)^n$$

where Erlang order = m-1 . The difficult problem of handling the gamma function (factorial) in the denominator of (2.20) in a numerical search is overcome by the manner in which the

"uncovered" gamma function is introduced as part of the numerical procedure. The "uncovered" gamma function, i.e., the gamma function in quadrature form, is used as a multiplier in (2.21) for the data moments rather than as a divisor for the quadrature moment expression as in (2.20).

This multiplier to the data, hereinafter called Modication factor and abbreviated MOD, is nothing other than the factorial or gamma function, so that

$$(2.21)\qquad \sum_{i=1}^{p} C_i t_i^m \left(\frac{t_i}{a}\right)^n = M_n \cdot m! = M_n \cdot MOD;$$

the MOD is the gamma function in quadrature form

$$(2.22)\qquad MOD = \sum_{i=1}^{p} C_i t_i^m = \lfloor(m+1) = m! \ ,$$

or its stepped-up equivalents as in (2.5) and applied in the same manner as in (2.10) and (2.23).

Thus, an Erlang of order m would appear in a system of non-linear equations together with the gamma function as

$$(2.23)\qquad \sum_{i=1}^{p} C_i t_i^m \left(\frac{t_i}{a}\right)^n = M_n \cdot \frac{1}{m+1} \sum_{i=1}^{q} C_i t_i^{m+1}$$

In the above expression, the summation indices are p and q on the left and right sides, respectively, to indicate that the number of quadrature points used on the left side and on the right side do not necessarily have to agree. In addition, on the right side, the gamma function is shown with a step-up of 1. Additional step-ups, more points, or both, will improve accuracy.

2.5 Alternate Weibull Form

Recall that the right hand side of (1.3) showed the closed form solution to the Weibull moments which, with $r=\frac{1}{b}$ will appear as

$$(2.24)\qquad M_n = \left(\frac{1}{a}\right)^{nr} \lfloor(1+nr) \ ,$$

and, since r is real, so is the product nr .

Using the MOD which is as valid for real arguments as it is for integer arguments, the following expression becomes available for use in a system of non-linear equations.

$$(2.25)\qquad \left(\frac{1}{a}\right)^{nr} = M_n \ / \ MOD,$$

where

$$(2.26)\qquad MOD = \sum_{i=1}^{p} C_i t_i^{nr} = \frac{1}{nr+1} \sum_{i=1}^{p} C_i t_i^{nr+1} = \lfloor(1+nr)$$

with a single step-up as above or with a higher order step-up.

This form uses quadrature to uncover the argument in the the gamma function $\lfloor(1+nr)$ in a manner almost identical to $\lfloor(m+1)$ of (2.22). The product nr is unknown since the shape parameter r is unknown; n is known because it is the particular number of the moment specified. In (2.22), m is the unknown order parameter.

3.0 CANONICAL FORMS AND NOTATION

3.1 Full Moment Canonical Form

The following quadrature canonical forms are presented for the exponential family of density functions including the Rayleigh. The identical approach can be used for the chi-square, hence for the normal as a by-product.

In addition, the full form is shown separately from the finite interval form; again, these could be combined but are shown and discussed separately for clarity.

The truncated model removes the limitation of the usual (0,oo) interval for Gauss-Laguerre quadrature by providing a method for the tails (T,oo) and intervals(T_1,T_2) and (0,T).

In addition, the subtlety pointed out earlier regarding accuracy when fractional powers are encountered is extended to the canonical form with its plethora of parameters, whereas, in the earlier sections, the application was to the simple gamma function.

The Full Moment Canonical Form is

(3.1)

$$M_n = \frac{a^{Q+m}}{m!(m+nr+1)(m+nr+2)..(m+nr+Q)} \cdot \sum_{i=1}^{p} C_i \left(\frac{t_i}{a}\right)^{m+nr+Q}$$

3.2 Finite Interval Canonical Form

The following, which, at first, seems to be limited to the interval (T,oo) will be found to be applicable to (0,T) as will be shown hence.

(3.2)

$$M_{nt} = -\frac{e^{-aT}}{m!} \sum_{Q=1}^{k} \frac{a^{Q+m} T^{m+nr+Q}}{(m+nr+1)(m+nr+2)..(m+nr+Q)} + \frac{e^{-aT}}{m!} \sum_{i=1}^{p} \frac{a^{Q+m} C_i \left(\frac{t_i}{a} + T\right)^{m+nr+Q}}{(m+nr+1)(m+nr+2)..(m+nr+Q)} \Bigg|_{Q=k}$$

3.2.1 Notation

The notation used is as follows:

Nomenclature:

- M_n — n^{th} Full Moment about origin, M_o = CDF
- M_{ns} — n^{th} Truncated Moment (0,T)
- M_{nt} — n^{th} Tail Moment (T,oo)
- m — Erlang order minus one, i.e. order=m+1; m=0 for exponential (Erlang order=1); m=0 for Weibull
- a — Parameter related to time constant
- r — Inverse of Weibull shape parameter b,

r=1/b
r=1 for Exponential or Erlang
r=.5 for Rayleigh,

p Number of quadrature points

C_i, t_i Christoffel numbers and knots

Q Step-up index
Q is integer, limited to
$Q < 2p-nr-1-m$

k Order of step-up

$\hat{T}$ Real-time

$T=\hat{T}$ for gamma, exponential, and Erlang

$T=\hat{T}^b$ for Weibull

T Truncation point, $\int_0^T$ or $\int_T^\infty$

T_1, T_2 Finite Interval, $\int_{T_1}^\infty - \int_{T_2}^\infty$

3.3 Finite Interval Quadrature - Derivation

In a manner similar to the two changes of variable used to obtain the full moment quadrature form, the finite interval quadrature requires two as well, but the second change of variable introduces the finite time "T".

The terminology will be consistent with the canonical form in that,

(3.3) $M_{ns} + M_{nt} = M_n$,

whence M_{ns} is the truncated(short) mean and M_{nt} is the mean of the tail.

There are two forms for the truncated mean. The first is the series expression discussed in connection with the canonical form, the second form uses quadrature without the series term. It is this second form which will be discussed now.

First, the density function and the moments of the Weibull are cast into the following form by the change of variable $u=x^b$, so that $T=T^b$ in accordance with the notation in the nomenclature section

(3.4) $f(x) = abx^{b-1}e^{-ax^b}dx$ becomes

$$f(u) = ae^{-au}du$$

(3.5) $M_{nt} = \int_T^\infty abx^{b-1}e^{-ax^b}x^n dx$ becomes

$$f(u) = \int_T^\infty ae^{-au}u^{(n/b)}du$$

Using a second change of variable, $t=a(u-T)$, thence $u = \frac{t}{a} + T$, and using r=1/b for convenience, the moments become,

(3.6) $$M_{nt} = \int_T^\infty ae^{-au}u^{nr}du$$

$$= \int_0^\infty e^{-a\left(\frac{t}{a}+T\right)}\left(\frac{t}{a}+T\right)^{nr}dt$$

$$= e^{-aT}\int_0^\infty e^{-t}\left(\frac{t}{a}+T\right)^{nr}dt$$

$$= e^{-aT}\sum_{i=1}^{p} C_i\left(\frac{t_i}{a}+T\right)^{nr}$$

and $$\sum_{i=1}^{p} C_i\left(\frac{t_i}{a^r}\right)^n = M_n \cdot MOD1$$

where

$$MOD1 = \frac{\Gamma(n+1)}{MOD} = \frac{n!}{(nr)!}$$

and MOD is as in (2.22) with nr replacing m so that

(3.7) $$\sum_{i=1}^{p} C_i\left(\frac{t_i}{a^r}\right)^n = (M_{ns}+M_{nt})MOD1 ,$$

thus

(3.8) $$\sum_{i=1}^{p} C_i\left(\frac{t_i}{a^r}\right)^n$$

$$= M_{ns} + e^{-aT}\sum_{i=1}^{p} C_i\left(\frac{t_i}{a}+T\right)^{nr} \quad MOD1$$

If dealing with reliability, M_{ns} are the truncated data moments with observations terminating at time "T" with M_{os} being the zeroth moment, the CDF.

The same result will be obtained by the alternate quadrature form,

(3.9) $$M_n = \sum_{i=1}^{p} C_i\left(\frac{t_i}{a}\right)^{nr}$$

$$= M_{ns} + e^{-aT}\sum_{i=1}^{p} C_i\left(\frac{t_i}{a}+T\right)^{nr}$$

So, for truncated data, two quadrature forms and one series form is available.

4.0 INITIAL CONDITION PROBLEM

4.1 Inconsistent Initial Conditions

The quadrature systems developed herein have the unknown variables embedded as arguments of exponentials. The argument can contain two unknowns, one a power of the other. During a search, one variable going negative raised to a non-integer power, which is the second variable, will terminate the search since the result would be an imaginary number. The immediate response

to this problem might be to prevent the offending variable from going negative during the search. Unfortunately, this also destroys the integrity of the rate-of-change vector matrix. Also, convergence to the correct result depends upon the exponential argument remaining negative during the search. The value of the exponential expression blows-up when the argument becomes positive and convergence is never achieved.

If either parameter is chosen inconsistent with respect to the other, the algorithm wanders indefinitely and never converges, or terminates when one variable becomes negative. With a choice of consistent parameters, but initially too far removed from the correct solution, again the algorithm will not converge.

Therefore, to be useful, a method must be introduced to choose properly the initial conditions.

A consistent initial condition is one which relates the unknown variables by a basic relationship applicable to the particular distribution being modelled. For example, the CDF relationship for the Weibull, $M_o=1-e^{-ax^b}$ relates the parameters a and b, given knowledge of M_o which is the fraction of units failed in time X. Therefore, an unrealistic initial value of b could be selected together with a consistent value of a computed by the CDF relationship.

Hence, two chores must be accomplished simultaneously, that of realistic and of consistent initial conditions.

4.2 Dominant Ratios Method, Weibull, for Approximating Initial Conditions

The Weibull truncated series is,

$$(4.1)\quad M_n(0,T) = \sum_{Q=1}^{k} \frac{T^{nr+Q} a^Q e^{-aT}}{(nr+1)(nr+2)..(nr+Q)} .$$

The first term of the series is dominant. For any moment, the first term has the same relative degree of inaccuracy; therefore, the ratio of the first terms is much more accurate than the values of the first terms themselves; i.e., M_1/M_o, and M_2/M_1.

The first terms are, for M_o, M_1, and M_2

$$(4.2)\quad M_o(0,T) = \frac{e^{-aT}(aT^1)}{1}$$

$$(4.3)\quad M_1(0,T) = \frac{e^{-aT}(aT^{r+1})}{r+1}$$

$$(4.4)\quad M_2(0,T) = \frac{e^{-aT}(aT^{2r+1})}{2r+1}$$

so that

$$(4.5)\quad M_1/M_o = \frac{e^{-aT}(aT^{r+1})/(r+1)}{e^{-aT}aT} = \frac{T^r}{r+1} .$$

With $X=T^b$ and $r=1/b$, where X is real-time, we find

$$(4.6)\quad M_1/M_o = \frac{X}{r+1} .$$

Solving for r,

$$(4.7)\quad r = X\frac{M_o}{M_1} - 1 .$$

Thus, the parameter r for the Weibull is found: it is time-dependent as well as a function of M_o and M_1. This provides a very good initial estimate of the Weibull parameter b since $b=1/r$.

With knowledge of a realistic initial value of b, a consistent value of a can be found by the CDF relationship, $M_o=1-e^{-ax^b}$. Solving for a

$$(4.8)\quad a = -\mathcal{L}n(1-M_o)/x^b$$

It can be shown that the series of (4.1), when taken to an infinite number of terms, is precisely the CDF; that is, for $n=0$, (4.1) is

$$(4.9)\quad CDF = M_o(0,T) = \sum_{Q=1}^{\infty} \frac{T^Q a^Q e^{-aT}}{Q!}$$

$$= e^{-aT}\left[\frac{-(aT)^0}{0!} + \sum_{Q=0}^{\infty}\frac{(aT)^Q}{Q!}\right]$$

$$= e^{-aT}(-1 + e^{aT}) = 1 - e^{-aT}.$$

The ratio M_2/M_1 yields

$$(4.14)\quad \frac{M_2}{M_1} = \frac{T^r(r+1)}{(2r+1)} = \frac{X(r+1)}{(2r+1)}$$

so that

$$(4.15)\quad r = \frac{XM_1 - M_2}{2M_2 - XM_1}$$

which is simple arithmetic.

References

References indicated in text by number in brackets, [ ]

1. Beckmann, P., Orthogonal Polynomials for Engineers and Physicists, Boulder: The Golem Press, 1973

2. Reddick, H.W., and Miller, F.H., Advanced Mathematics for Engineers, New York: John Wiley & Sons, Inc., 1938.

THE PROBABILITY INTEGRALS OF THE MULTIVARIATE NORMAL:

THE 2^n TREE AND THE ASSOCIATION MODELS

Dror Rom, Biometrics Research, Merck Sharp & Dohme

Sanat K. Sarkar, Temple University

1. Introduction

The standard multivariate normal density has the following form:

$$f(\mathbf{z}) = (2\pi|\Sigma|)^{-\frac{1}{2}} \exp(-\frac{1}{2}\mathbf{z}'\Sigma^{-1}\mathbf{z})$$

where

$$\Sigma = \begin{pmatrix} 1 & \rho_{12} & \dots & \rho_{1n} \\ \rho_{21} & 1 & \dots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & .. & 1 \end{pmatrix}$$

The evaluation of the probability integrals of the multivariate normal distribution is of great importance to statisticians. The joint distribution of several random variables is often assumed to be multivariate normal. This distribution also provides an approximation to many multivariate distributions including the multinomial distribution, when the sample size is large. Most of the work so far concentrated on either one of three cases: The evaluation of the bivariate and trivariate normal probability integrals; the evaluation of the multivariate normal probability integrals for special cases of the correlation matrix; the evaluation of the multivariate normal probability integrals for special domains. Not much work was done to achieve a resonable technique for the evaluation of the probability integarls of a general multivariate normal distribution on any (rectangular) region.

The techniques for the evaluation of the multivariate normal probabilities can be categorized as:

(1) Expansions of the density in power series.

(2) Reduction to lower dimensions and then using quadratures.

(3) Modeling the probability surface (log-linear models for example).

(4) Monte-Carlo integration techniques.

2. The Contingency Tables And Association Models.

Goodman (1981) developed the following association model: For an $I \times J$ contingency table, let F_{ij} denote the expected frequency in the ith row and jth column of the table $(i = 1,\dots,I; j = 1,\dots,J)$. Consider the following model for the expected frequencies

$$F_{ij} = \alpha_i \beta_j e^{(\phi \mu_i \nu_j)}. \qquad (1)$$

where α_i, β_j ,μ_i, ν_j and ϕ are parameters. Let Θ_{ij} denote the local cross product ratios given by

$$\Theta_{ij} = (F_{ij}F_{i+1,j+1})/(F_{i,j+1}F_{i+1,j}). \qquad (2)$$

$$(i = 1,\dots,I-1; j = 1,\dots,J-1)$$

From (1) and (2) we obtain

$$\gamma_{ij} = log\Theta_{ij} = \phi(\mu_i - \mu_{i+1})(\nu_j - \nu_{j+1}). \qquad (3)$$

If we let $\mu_i - \mu_{i+1} = \Delta$ $(i = 1, \ldots, I-1)$, $\nu_j - \nu_{j+1} = \Delta'$, $(j = 1, \ldots, J-1)$, (where Δ and Δ' are unspecified) we obtain the uniform association model. Holland and Wang (1987) investigated the extension of γ_{ij} to continuous bivariate densities $f(x,y)$. They showed that the limiting case of γ_{ij} is the bivariate function

$$\gamma_f(x,y)\Delta x \Delta y = \frac{\partial^2}{\partial x \partial y} log\, f(x,y)\, \Delta x \Delta y. \qquad (4)$$

They called $\gamma_f(x,y)$ the local dependence function for $f(x,y)$. For the bivariate normal density, the local dependence function is $\gamma_f(x,y) = \rho/(1-\rho^2)$. The logarithm of the local cross product ratio for the four infinitesimal rectangular region around (x,y), $(x, y+b)$ $(x+a, y)$ and $(x+a, y+b)$ is

$$\{\rho/(1-\rho^2)\}ab. \qquad (5)$$

which is independent of (x,y). Let

$$\phi = \bar{\rho}/(1-\bar{\rho}^2)$$

where

$$\bar{\rho} = \rho\{(1 - \delta_\mu{}^2/12)(1 - \delta_\nu{}^2/12)\}^{1/2}. \qquad (6)$$

and δ_μ and δ_ν are the widths of the corresponding row and column categories. Note that (6) is Sheppard's correction for grouped data (Kendall and Stuart (1976)). Model (1) becomes:

$$F_{ij} = \alpha_i \beta_j e^{\mu_i \nu_j \frac{\bar{\rho}}{(1-\bar{\rho}^2)}}. \qquad (7)$$

In this form, α_i and β_j can be looked at as main effects in the model, and μ_i and ν_j are the centers of the ith row and jth column respectively. Wang (1987) showed that (7) has approximately the same local cross product ratios as does the bivariate normal density. By Theorem 2.1.1 of Wang (1987), if model (7) is used to approximate the bivariate normal probabilities, then, if the resulting contingency table will have marginals which are univariate normal probabilities, the cell frequencies will fit well the bivariate normal probabilities. However, since in general, α_i and β_j are not known, model (7) can not be used directly. Hence both Goodman (1981) and Wang (1987) use the proportional fiting algorithm to obtain the cell frequencies. Noticing That if we drop α_i and β_j from model (7), the cell frequencies will still have the same local cross product ratios as does the corresponding bivariate normal density, we can use

$$e^{\{\mu_i \nu_j \frac{\bar{\rho}}{(1-\bar{\rho}^2)}\}}$$

as starting values for the proprtional fiting algorithm. This procedure cycles alternately between row scalings and column scalings until both row totals and column totals have been matched. Bishop, Fienberg & Holland (1975) showed that for complete two-way table this algorithm always converges.

Drawbacks of The Proportional Fitting Algorithm.

The contingency table approach, although provides an interesting application of the proportional fiting algorithm in improving the existing methods for computing bivariate normal probabilities, has some major drawbacks. Among them are the following:

(1) A preliminary set of the univariate normal probabilities is needed.

(2) An undetermined number of iterations until convergence. This is shown in Wang (1987). The number of iterations until convergence is 3 when ρ is 0.05, while the number of iterations until convergence is 30 when ρ is 0.95.

(3) The procedure always generates a full contingency table since it has to match the marginal probabilities with the univariate normal. This requires a substantial number of computations even when the probability over a small rectangle is needed.

(4) The procedure requires a lot of memory space to store all cell probabilities, especially as the dimensionality increases.

(5) It is not easily extended to higher dimensions.

In the next section we will show that these drawbacks can be overcome by utilizing a general linear model together with the 2^n Tree technique.

3. Approximating The Main Effects In The Log-Linear Model.

Following Goodman's (1981) log linear model for the bivariate normal density as presented in (1), we consider now a different way to approximate the bivariate normal probabilities. Since the standard bivariate normal density is symmetric in its arguments, it would be natural to assume that the main effects in (1) are equal, i.e., $\alpha_i = \beta_i$, $i = 1, \ldots, I$, and $\mu_i = \nu_i$, $i = 1, \ldots, I$.

Suppose for the moment that the diagonal elements of the contingency table F_{ij} are known. Then from (7) we obtain:

$$F_{ii} = {\alpha_i}^2 e^{{\mu_i}^2 \frac{\bar{\rho}}{1-\bar{\rho}^2}}. \tag{8}$$

Which holds for all $i = 1, \ldots, I$. From (1) we obtain:

$$\alpha_i = e^{\frac{1}{2}(\log F_{ij} - {\mu_i}^2 \frac{\bar{\rho}}{1-\bar{\rho}^2})}. \tag{9}$$

(9) provides a way to estimate the main effects as functions of the corresponding diagonal elements. This in turn gives a method for computing all F_{ij} elements provided the diagonal elements are known. This is done by estimating the main effects and then re-substituting them in (1).

4. Computing The Diagonal Elements of a Contingency Table.

Consider a two dimensional integrable function $f(x_1, x_2)$ on a closed interval $[a_1, b_1] \times [a_2, b_2]$. Suppose we want to evaluate the integral

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x_1, x_2) \partial x_1 \partial x_2. \tag{10}$$

We propose the following procedure to approximate (8)

Step 1

Let

$$I_0 = \frac{(b_1 - a_1)(b_2 - a_2)}{12} \big(f(a_1, a_2) + f(a_1, b_2) + f(b_1, a_1) + f(b_1, b_2) + 8f(a_0, b_0)\big). \tag{11}$$

where

$$(a_0, b_0) = \left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}\right)$$

Step 2

Partition $[a_1, b_1] \times [a_2, b_2]$ (the node) into four equally spaced rectangles (children) using (a_0, b_0) as a pivotal point. Use (11) to approximate the integral over each of the resulting rectangles. Let I_1 denote the sum of integrals over all four children. Let δ be a convergence criterion, then if $\epsilon = |I_1 - I_0| \leq \delta$ deliver I_1 as the approximated integral for (10), otherwise go to Step 3.

Step 3

Apply Step 1 and Step 2 to each child sequentially were the convergence criterion over each child is $\delta' = \frac{\delta}{4}$.

The quadrature in (11) is designed to fit perfectly a

second degree polynomial, i.e., if $f(x_1, x_2)$ is a second degree polynomial, then (11) will be exact.

The above is a recursive algorithm. Its main property is that it concentrates on regions where the function is not well behaved and spends less time elsewhere.

5. The Error In The 2^n Tree.

To compute the error, we can without loss of generality, assume that

$$[a_1, b_1] \times [a_2, b_2] = [-a, a] \times [-b, b]$$

and

$$(a_0, b_0) = (0, 0)$$

Now, we want the error

$$\int_{-a}^{a}\int_{-b}^{b} f(x_1, x_2)\partial x_1 \partial x_2 - \frac{4ab}{12}\big(f(-a, a) + f(-a, b) + f(a, -b) + f(a, b) + 8f(0, 0)\big). \tag{12}$$

We expand $f(x, y)$ in Tailor series around $(0, 0)$ and substitute throughout (12). All first and second order terms vanish as they should for they constitute a second degree polynomial. The third order term in the expansion vanishes by the virtue of odd symmetry. Hence we have (after integrating the fourth order term):

$$Error = \frac{-ab^5}{45}\frac{\partial^4 f(x_1, x_2)}{\partial {x_1}^4} - \frac{ba^5}{45}\frac{\partial^4 f(x_1, x_2)}{\partial {x_2}^4} - \frac{2a^3b^3}{9}\frac{\partial^4 f(x_1, x_2)}{\partial {x_1}^2 \partial {x_2}^2} + \cdots \tag{13}$$

Equation (13) gives the error when applying the quadrature (11) to the region $[-a, a] \times [-b, b]$.

6. The Log-Linear Model And The 2^n Tree.

We now combine the 2^n-Tree and the log-linear model as follows. Each of the diagonal elements (nodes) is partitioned into four equally spaced rectangles (children). The diagonal elements are computed by the above quadrature whereas the off-diagonal elements are computed using the log-linear model. We then combine the area over the children and compare with the node. If convergence has been reached, we stop partitioning this element, otherwise we partition each of the diagonal elements and apply the same procedure. This technique besides being fast, requires little memory space since only one diagonal element is examined at a time. This property is extremely useful in higher dimensions.

7. The Trivariate Standard Normal Distribution

Extending (1) to the trivariate normal distribution, we put the following model for the expected frequencies F_{ijk}

$$\log F_{ijk} = (1 - {\rho_{23}}^2)\delta_i^A + (1 - {\rho_{13}}^2)\delta_j^B + (1 - {\rho_{12}}^2)\delta_k^C + \frac{1}{\Delta}\Big((\rho_{13}\rho_{23} - \rho_{12})\mu_i^A\mu_j^B + (\rho_{12}\rho_{23} - \rho_{13})\mu_i^A\mu_k^C + (\rho_{12}\rho_{13} - \rho_{23})\mu_j^B\mu_k^C\Big) \tag{14}$$

where $$\Delta = \det\begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{pmatrix}.$$

We assume that

$$\delta_i^A = \delta_i^B = \delta_i^C = \delta_i \quad and \quad \mu_i^A = \mu_i^B = \mu_i^C = \mu_i \qquad \forall i.$$

Putting model (14) for the diagonal elements and solving for δ_i we get

$$\delta_i = \frac{(\log F_{iii} - \mu_i^2\Phi)}{3 - (\rho_{12}^2 + \rho_{13}^2 + \rho_{23}^2)}. \tag{15}$$

where

$$\Phi = -\frac{1}{\Delta}\left(\rho_{12}\rho_{13} + \rho_{12}\rho_{23} + \rho_{13}\rho_{23} - (\rho_{12} + \rho_{13} + \rho_{23})\right)$$

As for the bivariate normal distribution, we can solve for the main effects provided the diagonal elements are known. We again use the 2^n Tree combined with the linear model to generate all diagonal elements.

8. Comparisons

We compare here the results obtained from the log-linear models with some known results. For the bivariate normal probabilities, we compare with tabulated probabilities given by Goodman (1981). For the trivariate normal probabilities, we make this comparison with the tabulated probabilities given by Gupta (1963) and the exact results for some special cases of the domain of integration. These exact results were given by David (1953).

Table 1

Probabilities under the bivariate normal density with $\rho = 0.5$. First entry is the tabulated probability. Second entry is the computed probability. The computation were carried out in a *single precision* with maximum error less than 0.0001.

	x_1=0.0	x_1=0.5	x_1=1.0	x_1=1.5	x_1=2.0	x_1=2.5
	6.00	11.00	14.00	13.00	9.00	7.00
	5.97	10.60	13.72	12.93	8.92	6.70
x_2=2.5						
	23.00	35.00	37.00	29.00	16.00	9.00
	23.47	34.59	36.95	28.59	16.04	8.92
x_2=2.0						
	79.00	99.00	90.00	60.00	29.00	13.00
	78.69	99.04	90.31	59.69	28.59	12.93
x_2=1.5						
	191.00	205.00	160.00	90.00	37.00	14.00
	191.11	205.42	160.00	90.31	36.95	13.72
x_2=1.0						
	336.00	309.00	205.00	99.00	35.00	11.00
	336.32	308.75	205.42	99.04	34.59	10.60
x_2=0.5						
	429.00	336.00	191.00	79.00	23.00	6.00
	428.89	336.32	191.11	78.69	23.47	5.97
x_2=0.0						
	396.00	266.00	129.00	45.00	12.00	2.00
	396.39	265.50	128.86	45.31	11.54	2.45
x_2=-0.5						
	266.00	152.00	63.00	19.00	4.00	1.00
	265.50	151.89	62.95	18.90	4.12	0.73
x_2=-1.00						
	129.00	63.00	22.00	6.00	1.00	0.00
	128.86	62.95	22.28	5.72	1.06	0.16
x_2=-1.5						
	45.00	19.00	6.00	1.00	0.00	0.00
	45.31	18.90	5.72	1.26	0.03	0.03
x_2=-2.0						
	12.00	4.00	1.00	0.00	0.00	0.00
	11.54	4.11	1.06	0.03	0.02	0.00
x_2=-2.5						
	2.00	1.00	0.00	0.00	0.00	0.00
	2.45	0.73	0.16	0.03	0.00	0.00

Note: Probability=entry/10000.

Table 2

Selected probabilities under the trivariate normal density. Results from the proposed technique are compared with tabulated probabilities (when available) given by Gupta (1963), and exact probabilities (when available). The computations were carried out in a single precision with maximum error less than 0.001.

Correlations	Interval			Gupta	Exact	Log-linear
0.5 0.5 0.5	[-∞,0]	[-∞,0]	[-∞,0]	0.25000	0.25000	0.24965
	[-∞,0]	[-∞,0]	[0,∞]		0.08333	0.08308
	[-∞,0]	[0,∞]	[-∞,0]		0.08333	0.08308
	[-∞,0]	[0,∞]	[0,∞]		0.08333	0.08308
	[0,∞]	[-∞,0]	[-∞,0]		0.08333	0.08307
	[0,∞]	[-∞,0]	[0,∞]		0.08333	0.08307
	[0,∞]	[0,∞]	[-∞,0]		0.08333	0.08307
	[0,∞]	[0,∞]	[0,∞]	0.25000	0.25000	0.24963
0.3 0.5 0.7	[-∞,0]	[-∞,0]	[-∞,0]		0.25262	0.25214
	[-∞,0]	[-∞,0]	[0,∞]		0.04588	0.04561
	[-∞,0]	[0,∞]	[-∞,0]		0.08072	0.08040
	[-∞,0]	[0,∞]	[0,∞]		0.12079	0.12044
	[0,∞]	[-∞,0]	[-∞,0]		0.12079	0.12044
	[0,∞]	[-∞,0]	[0,∞]		0.08072	0.08040
	[0,∞]	[0,∞]	[-∞,0]		0.04588	0.04561
	[0,∞]	[0,∞]	[0,∞]		0.25262	0.25214
0.5 0.5 0.5	[3,∞]	[3,∞]	[3,∞]	0.00002		0.00002
	[2,∞]	[2,∞]	[2,∞]	0.00137		0.00137
0.7 0.7 0.7	[2,∞]	[2,∞]	[2,∞]	*0.00389*		0.00389
	[1,∞]	[1,∞]	[1,∞]	0.05756		0.05746
	[0,∞]	[0,∞]	[0,∞]	0.31011	0.31011	0.30948

REFERENCES

Bishop, Y. M. M., Fienberg, S. E. and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. Cambridge, Mass: MIT Press.

David, F. N. (1953). A note on the evaluation of the multivariate normal integral. *Biometrika* **40** 458-459.

Goodman, L. A. (1981). Association Models and the bivariate normal for contingency tables with ordered categories. *Biometrika* **68**, 347-55.

Gupta, S. S. (1963a). Probability integrals of the multivariate normal and multivariate t.*Ann. Math. Statist.***34**, 792-828.

Gupta, S. S. (1963b). Bibliography on the multivariate normal integral and related topics.*Ann. Math. Statist.***34**, 792-828.

Holland, P. W. and Wang, Y. J. (1987). Dependence functions for continuous bivariate densities. *Comm. Statist., Theory Meth.* To appear.

Kendall, M. G. and Stuart, A. (1976). *The Advanced Theory of Statistics*,**1**, 4^{th} ed. London: Griffin.

Wang, Y. J. (1987). The probability integrals of bivariate normal distributions: a contingency table approach. *Biometrika* **74**, 1985-1990.

XI. STATISTICAL METHODS

Multiple–Smoothing Parameters in Semiparametric Multivariate Model Building
Grace Wahba, University of Wisconsin–Madison

Computing Empirical Likelihoods
Art Owen, Stanford University

Computing Extended Maximum Likelihood Estimates for Linear Parameter Models
Douglas B. Clarkson, IMSL, Inc.; Robert I. Jennrich, UCLA

Simultaneous Confidence Intervals in the General Linear Model
Jason C. Hsu, Ohio State University

Assessment of Prediction Procedures in Multiple Regression Analysis
Victor Kipnis, University of Southern California

Posterior Influence Plots
Robert E. Weiss, University of Minnesota

Exact Power Calculations for the Chi–Square Test of Two Proportions
Carl E. Pierchala, U.S. Department of Agriculture

On Covariances of Marginally Adjusted Data
James S. Weber, Roosevelt University

Optimizing Linear Functions of Random Variables Having a Joint Multinomial or Multivariate Normal Distribution
J.P. De Los Reyes, University of Akron

Approaches for Empirical Bayes Confidence Intervals for a Vector of Exponential Scale Parameters
Bradley P. Carlin, Alan E. Gelfand, University of Connecticut

A Data Analysis and Bayesian Framework for Errors–in–Variables
John H. Herbert, U.S. Department of Energy

The Effect of Low Covariate–Criterion Correlations on the Analysis–of–Covariance
Michael J. Rovine, Alexander von Eye, Phillip Wood, Pennsylvania State University

Estimation of the Variance Matrix for Maximum Likelihood Parameters Using Quasi–Newton Methods
Linda Williams Pickle, National Cancer Institute; Garth P. McCormick, George Washington University

Application of Posterior Approximation Techniques to the Ordered Dirichlet Distribution
Thomas A. Mazzuchi, Refik Soyer, George Washington University

Comparison of "Local Model" Statistical Classification Methods
Daniel Normolle, University of Michigan

An Example of the Use of a Bayesian Interpretation of MDA Results
James R. Nolan, Siena College

Unbiased Estimates of Multivariate General Moment Functions of the Population and Application to Sampling Without Replacement from a Finite Population
Nabih N. Mikhail, Liberty University

MULTIPLE SMOOTHING PARAMETERS IN SEMIPARAMETRIC MULTIVARIATE MODEL BUILDING

Grace Wahba
University of Wisconsin-Madison

1. INTRODUCTION

Semiparametric model building, particularly using multivariate splines of various types, has the potential to allow the organization and analysis of large data sets which represent responses as a function of several variables. One of the major stumbling blocks to the further development of these techniques has been the heavy and sometimes prohibitive computational cost of estimating multiple smoothing parameters. Some recent work in Madison (Gu et al. (June 1988), Gu (June 1988)) has resulted in improved numerical methods for speeding up the calculation of both GCV (generalized cross validation) and GML (generalized maximum likelihood) estimates of multiple smoothing parameters. (See Wahba (1985) and references there for a discussion of these estimates.)

This work has allowed us to explore the use of interaction smoothing splines (Barry (1986), Wahba (1986),Gu et al. (June 1988)) for multivariate exploratory model building, and also to tackle some interesting problems concerning the merging of data from different sources with different and only partially known error structures.

In the first part of this paper we will briefly discuss the data-merging problem and in the second part we will describe some recent model building work with interaction smoothing splines with multiple smoothing parameters. We note that J. Friedman's extremely interesting keynote talk at this conference concerned what might be called interaction regression splines. There are philosophically interesting similarities as well as contrasts in his approach and the one we describe.

1. A DATA MERGING PROBLEM ARISING IN METEOROLOGY

To motivate the problem, we first describe a very special concrete case, then we consider a more general version. Further details appear in Wahba (1988).

Let P be (latitude, longitude) and let $f(P)$ be the 500 millibar height, that is, the height in the atmosphere at which the pressure is 500 millibars. Every 12 hours the global radiosonde (weather balloon) network observes the 500 mb height and reports the obervations.

$$y_i^{(0)} = f(P_i) + \varepsilon_i^{(0)} \quad ,i=1,\ldots,n$$

where the $\varepsilon_i^{(0)}$ are treated as independent random errors with a common variance σ_0^2. Here we will treat all random variables as Gaussian with 0 mean unless otherwise specified. Simultaneously, there is a forecast of the state variables of the model, which can be converted to a forecast of $f(P)$. Let this forecast be

$$y_i^{(f)} = f(P_i) + \varepsilon_i^{(f)}$$

where $e_i^{(f)}$ is the forecast error. The problem is to merge the observational and forecast data to get a new estimate for the 500 mb height, which is then used as part of the initial conditions for a numerical weather forecast model. Generally, the error variance of the observations depends on the equipment and is known. The forecast error is generally correlated, and depends on the particular forecast model in question. It has not been as well known as one would wish. Recent work at the European Center for Medium Range Forecasts (ECMWF) (Hollingsworth and Lonnberg (1986), Lonnberg and Hollingsworth (1986)) has provided some fairly detailed information on the 500mb forecast error covariance structure of the model there. The forecast error spatial covariance was estimated, based on analysis of three months data comparing observation to forecast. Use of these results can be used to "retune" the estimation of initial conditions (which is then of course going to change the forecast error covariance.) The forecast error can depend on many things, including the weather itself, and we were interested in seeing wether useful information concerning forecast error covariance could be obtained dynamically, that is, from one instantaneous set of 500 millibar height observations,which consist of the order of 600-1000 observations, and one global forecast. If it can, then the information can be fed back into the model, to improve the estimation of the initial conditions. In Wahba (1988) the forecast error covariance is modelled by

$$E\varepsilon_i^{(f)}\varepsilon_j^{(f)} = \sigma_f^2 Q_\theta(P_i,P_j),$$

where $Q_\theta(\cdot,\cdot)$ is an isotropic correlation function on the sphere depending on the single parameter θ and defined by

$$Q_\theta(P_i,P_j) = \rho_\theta(\gamma(P_i,P_j))$$

$$\rho_\theta(\gamma) = \frac{(1-2\theta\cos\gamma+\theta^2)^{-1/2}-(1+\theta)^{-1}}{(1-\theta)^{-1}-(1+\theta)^{-1}}$$

where γ is the angular distance between P_i and P_j.

Figure 1 is a plot of $\rho_\theta(\gamma)$ for seven values of θ. θ is a monotone function of the 1/2 power point $\gamma_{1/2}$ (the distance for which the correlation is down to 1/2), and $\gamma_{1/2}$ is probably the single most important parameter (in a practical sense) of an isotropic correlation function on the sphere. This family of correlation functions was chosen for mathematical convenience and for the resemblance of some of its members to correlation functions in Hollingsworth and Lonnberg (1986), Lonnberg and Hollingsworth (1986).

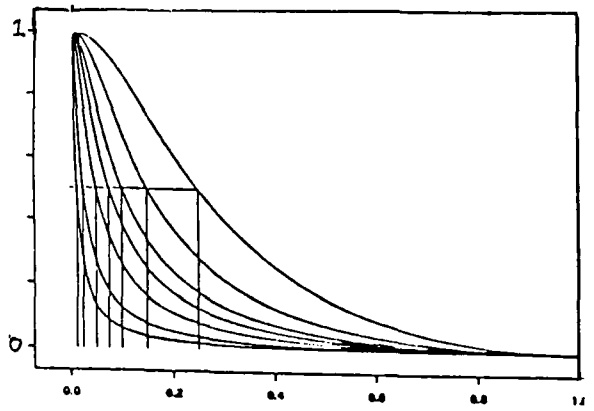


Fig. 1. $\rho_\theta(\gamma)$, *for seven different values of* θ.

Making the reasonable assumption that the $\varepsilon_i^{(o)}$ and $\varepsilon_i^{(f)}$ are independent, then one can estimate σ_f^2, $r = \dfrac{\sigma_o^2}{\sigma_f^2}$ and θ by maximum likelihood, by considering

$$z_i = y_i^{(o)} - y_i^{(f)},$$

then $z=(z_1, \ldots, z_n)'$ has the distribution

$$z \sim N(0, \sigma_f^2(rI + Q_\theta))$$

where Q_θ is the $n \times n$ matrix with i,jth entry $Q_\theta(P_i, P_j)$.

The maximum likelikhood estimates of r and θ can be shown to be the minimzers of

$$M(r,\theta) = \frac{z'(rI + Q_\theta)^{-1} z}{[det(rI + Q_\theta)^{-1}]^{\frac{1}{n}}}$$

and the ML estimate of σ_f^2 is

$$\hat{\sigma}_f^2 = \frac{1}{n} z'(\hat{rI} + Q_{\hat{\theta}})^{-1} z$$

where $\hat{r}$ and $\hat{\theta}$ are the ML estimates of r and θ.

An efficient algorithm suitable for minimizing $M(r, \theta)$ (as well as the GCV function, to be discussed later) with large data sets has been proposed by Gu et al. (June 1988), and some code is available in Gu (June 1988). An outline of the algorithm goes as follows:

i) For fixed θ, tridiagonalize Q_θ as

$$U^T Q_\theta U = T_\theta$$

where U is orthogonal and T is tridiagonal, by successively applying the Householder transformation. A strategy for speeding up this step by appropriate truncation which sets suitably small elements of the diagonal of T_θ to 0 appears in Gu et al. (June 1988) and is in the code in Gu (June 1988). Then

$$M(r,\theta) = \frac{h'(rI + T_\theta)^{-1} h}{[det(rI + T_\theta)^{-1}]^{\frac{1}{n}}}$$

where $h=Uz$.

ii) For each trial value of r, do a Cholesky decomposition $C'C$ of $(rI + T_\theta)$, where C is upper bidiagonal,

$$C = \begin{bmatrix} a_1 & b_1 & & & \\ & a_2 & b_2 & & \\ & & \ddots & & \\ & & & a_{n-1} & b_{n-1} \\ & & & & a_n \end{bmatrix}$$

iii) The numerator of M is then computed by back substitution and the denominator as $(\prod_{i=1}^{n} a_i)^{\frac{2}{n}}$.

iv) For fixed θ conduct a search in log$r$, then step to a new θ.

Most of the work is in the tridiagonalization, thus, a search is "cheap" in r and expensive in θ.

This algorithm has allowed us to ask practical questions based on realistic simulated data, with n=600 or more, using our Sun workstation. Such questions as, can r be estimated sufficiently accurately given one set of data, to make the method useful in a practical sense, for adjusting the relative weights to be given in observation and forecast when merging them to get new initial conditions. For example, Figure 2 gives a histogram of log$\hat{r}$ from a simulation experiment with 1000 replications. (Note that the matrix decompositions above are only done once!) For this simulation it was assumed that θ was known, and it was taken to correspond to a realistic value of the half power point of 500 km. Data was simulated for 611 Northern Hemisphere radiosonde stations witha true r of 2/3. Percentiles of the distribution are given on the plot, and one can see that, under the ideal conditions of the simulation, one could reliably detect a drop in r from the nominal 2/3 to .43 or less.

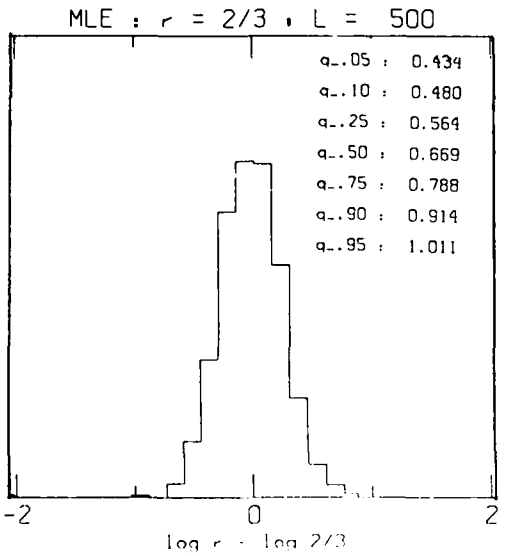


Fig. 2. Histogram for $\log\hat{r}$.

We remark that this problem of estimating relative accuracy generalizes to indirectly sensed data from different types of instruments. Let f be some meteorological field of interest, and suppose we have two different sources of data,

$$y^{(1)} = L^{(1)}f + \varepsilon^{(1)}$$

and

$$y^{(2)} = L^{(2)}f + \varepsilon^{(2)}$$

where

$$y^{(\alpha)} = (y_1^{(\alpha)}, \ldots, y_{n_{(\alpha)}}^{(\alpha)}),$$

$$L^{(\alpha)} = (L_1^{(\alpha)}, \ldots, L_{n_{(\alpha)}}^{(\alpha)}),$$

$$\varepsilon^{(\alpha)} = (\varepsilon_1^{(\alpha)}, \ldots, \varepsilon_{n_{(\alpha)}}^{(\alpha)}), \quad \alpha = 1,2.$$

If f were the three dimensional atmospheric temperature distribution one source could possibly be the important case of satellite observed radiances. Suppose that $\varepsilon^{(\alpha)} = N(0, \sigma_{(\alpha)}^2 \Sigma^{(\alpha)})$ and it is desired to estimate $r = \dfrac{\sigma_{(2)}^2}{\sigma_{(1)}^2}$, and, possibly , some parameters in $\Sigma^{(\alpha)}, \alpha=1,2$. To proceed as before, we need the existence of two matrices $B^{(\alpha)}, \alpha=1,2$ of dimension $n \times n_{(\alpha)}, \alpha=1,2$ for some sufficiently large n, which satisfy

$$B^{(1)}L^{(1)}f = B^{(2)}L^{(2)}f.$$

Let $w^{(\alpha)}$ be defined by

$$w^{(\alpha)} = B^{(\alpha)}y^{(\alpha)}, \quad \alpha=1,2$$

and let ξ be defined by

$$\xi = w^{(1)} - w^{(2)} = B^{(1)}y^{(1)} - B^{(2)}y^{(2)} = B^{(1)}\varepsilon^{(1)} - B^{(2)}\varepsilon^{(2)}.$$

The covariance matrix of ξ is then

$$E\xi\xi' = \sigma_1^2 B^{(1)} Q_1 B^{(1)\prime} + \sigma_2^2 B^{(2)} Q_2 B^{(2)\prime}.$$

Suppose $B^{(1)}Q_1B^{(1)\prime}$ is of full rank, then we can take the Cholesky decomposition LL' of $B^{(1)}Q_1B^{(1)}$, where L is lower triangular, and let $z = L^{-1}\xi$. Then the covariance matrix of z is

$$Ezz' = \sigma_2^2(rI + Q),$$

where $r = \sigma_1^2/\sigma_2^2$ and $Q = L^{-1}B^{(2)}Q_2B^{(2)}L^{-1\prime}$. The ML estimate is then given by the minimizer of M and the estimability of r depends on the properties of Q. Loosely speaking, the two vectors $w^{(1)}$ and $w^{(2)}$ need have to have their "energy" at different "wavenumbers". Questions about the existence of good, or consistent estimates, as $n \to \infty$ can be approached by studying the properties of Q from the point of view of the theory of equivalence and perpendicularity. See Stein (1988),Stein (1987). Wahba (1988), Wahba (March, 1987).

2. INTERACTION SPLINES

Interaction splines provide a tool for modelling a response which may depend nonparametrically on d variables, as sums of (smooth) functions of one variable, sums of functions of two variables, and so forth. Sums of functions of one variable are known as additive models, see, for example Friedman, Grosse, and Stuetzle (1983), Stone (1985), Hastie and Tibshirani (1987). Interaction splines loosely fall into two types, namely, regression splines, whereby the estimate is a least squares regression on a set of basis functions (the number and types of of which play the role of the smoothing parameter(s)), and smoothing splines, where the estimate is the solution of a penalized least squares problem in a reproducing kernel hilbert space with an appropriate norm or seminorm. (Such estimates are always Bayes estimates, see Kimeldorf and Wahba (1971), Wahba (1978).) Hybrid splines result when one solves the penalized least squares problem in a finite dimensional (approximating) space of basis functions. In this case, the multiplier(s) on the penalty(ies), and the number of basis functions may both act as smoothing parameters. J. Friedman, (these proceedings) was concerned with regression splines, in this Section we are concerned with smoothing splines. Computing the GCV estimate for the number of basis functions for regression splines is not a problem (altho modifications to the GCV to account for knot selection raise interesting questions, see Friedman (August 1988), Friedman and Silverman (1987)). The computation of the GCV function in the smoothing case can be a major numerical challenge when there are large data sets and multiple smoothing parameters, and heretofore has been a deterrent to work with multiple smoothing parameters.

The reproducing kernel (rk) hilbert space that we and others (Barry (1983), Barry (1986), Wahba (1986), Gu et al. (June 1988) have proposed as the natural setting for interaction smoothing splines, is the tensor product of d one dimensional rk spaces (see Wahba (1975) for an older work on tensor product spaces). We remark that the tensor product spline spaces are qualitatively different that the spaces which provide the setting for the thin plate splines (see Wahba and Wendelberger (1980) for example). Some remarks contrasting tensor product and thin plate splines may be found in Wahba (1986).

In the remainder of this paper, we describe interaction splines, and show how the algorithm proposed in Gu et al. (June 1988) can be used to choose multiple smoothing parameters and build interaction spline models via the use of GCV.

We first describe an abstract result concerning the fitting of functions with different smoothing parameters associated with different components of the estimate. The application to interaction spline models will then be fairly easy and will be described next.

Let $H = H_0 \oplus H_1$ be a reproducing kernel Hilbert space of functions of $x = (x_1, \ldots, x_d)$ where H_0 is of finite dimension M and H_1 is the direct sum of p orthogonal subspaces $H^1, \ldots, H^p$,

$$H_1 = \sum_{\beta=1}^{p} \oplus H^\beta. \tag{2.1}$$

Suppose we wish to find $f \in H$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x(i)))^2 + \lambda \sum_{\beta=1}^{p} \theta_\beta^{-1} \| P^\beta f \|^2 \tag{2.2}$$

where $x(i) = (x_1(i), \ldots, x_n(i))$, $\theta_1 = 1$ and P^β is the orthogonal projector in H onto H^β. (The H^β will later be various subspaces for main effects, two factor interactions, etc.) If the rk for H^β with squared norm $\| P^\beta f \|^2$ is $Q^\beta(\cdot\,;\cdot)$, then the rk for H^1 with squared norm $\sum_{\beta=1}^{p} \theta_\beta^{-1} \| P_\beta f \|^2$ is

$$\sum_{\beta=1}^{p} \theta_\beta Q^\beta(\cdot\,;\cdot) = Q_\theta(\cdot\,;\cdot), \tag{2.3}$$

say. The following facts are well known: (Kimeldorf and Wahba (1971), Wahba (1978)) Let $\phi_1, \ldots, \phi_M$ span H_0 and suppose the design points $x(1), \ldots, x(n)$ are such that least squares regression in H_0 is unique. Then the minimizer $f_{\lambda,\theta}$ of (2.2) is defined by

$$f_{\lambda,\theta}(x) = \sum_{\nu=1}^{M} d_\nu \phi_\nu(x) + \sum_{i=1}^{n} c_i Q_\theta(x; x(i))$$

where $d = (d_1, \ldots, d_M)$ and $c = (c_1, \ldots, c_n)$ satisfy

$$(Q_\theta + n\lambda I)c + Sd = y$$

$$S'c = 0$$

where Q_θ is the $n \times n$ matrix with ijth entry $Q_\theta(x(i)\,; x(j))$ and S is the $n \times M$ matrix with $i\nu$th entry $\phi_\nu(x(i))$. Letting the Q-R decomposition of S be

$$S = (F_1 : F_2)\begin{bmatrix} R \\ 0 \end{bmatrix}$$

a series of standard calculations (see e. g. Wahba (1985)) gives that the influence matrix $A(\lambda,\theta)$ for this problem is

$$I - A(\lambda,\theta) = n\lambda F_2 (F_2' Q_\theta F_2 + n\lambda I)^{-1} F_2'$$

and the GCV function becomes

$$V(\lambda,\theta) = \frac{z'(n\lambda I + \Sigma_\theta)^{-2} z}{(tr(n\lambda I + \Sigma_\theta)^{-1})^2} \tag{2.4}$$

where

$$z = F_2' y$$

and

$$\Sigma_\theta = \Sigma^1 + \theta_2 \Sigma^2 + \cdots + \theta_p \Sigma^p$$

with

$$\Sigma^\beta = F_2' Q^\beta F_2$$

and Q^β is the $n \times n$ matrix with ijth entry $Q^\beta(x(i)\,; x(j))$.

The algorithm for minimizing $V(\lambda,\theta)$ suggested in Gu et al. (June 1988) begins with steps i) and ii) as in Section 1. The numerator of V is obtained by backsubstitution, and to calculate the denominator we need to calculate $tr(C^{-1}C^{-1\prime})$. Denote the i-th row of C^{-1} by c_i'. We have $tr(C^{-1}\,C^{-1\prime}) = \sum \| c_i \|^2$. From

$$C^{-1\prime}C' = (c_1, c_2, \ldots, c_n)\begin{bmatrix} a_1 & & & & \\ b_1 & a_2 & & & \\ & b_2 & \ddots & & \\ & & & a_{n-1} & \\ & & & b_{n-1} & a_n \end{bmatrix} = I$$

we have

$$a_n c_n = e_n$$

$$a_i c_i = e_i - b_i c_{i+1}, \qquad i = n-1, \ldots, 1$$

where e_i's are unit vectors. Because $C^{-1\prime}$ is lower triangular, c_{i+1} is orthogonal to e_i. Thus we have the recursive formula

$$\| c_n \|^2 = a_n^{-2}$$

$$\| c_i \|^2 = (1 + b_i^2 \| c_{i+1} \|^2)\, a_i^{-2}, \qquad i = n-1, \ldots, 1$$

which can be calculated in $O(n)$ flops.

We now describe how one obtains interaction spline models. Let W_2^m be the Sobolev space

$$W_2^m = \{f: f, f', \ldots, f^{(m-1)} \text{ abs. cont.}, f^{(m)} \in L_2[0, 1]\}$$

with the squared norm

$$\|f\|^2_{W_2^m} = \sum_{\nu=0}^{m-1} (R_\nu f)^2 + \int_0^1 (f^{(m)}(x))^2 dx,$$

where

$$R_\nu f = \int_0^1 f^{(\nu)}(x)dx, \qquad \nu = 0, 1, \ldots, m-1.$$

Let $k_l(x) = B_l(x)/l!$, where B_l is the l-th Bernoulli polynomial (Abromowitz and Stegun (1965)), we have $R_\nu B_l = \delta_{\nu,l}$ where $\delta_i = 1$, $i=0$, and 0 otherwise. With this norm, W_2^m can be decomposed as the direct sum of m orthogonal one-dimensional subspaces $\{k_l\}$, $l = 0, 1, \ldots, m-1$, where $\{k_l\}$ is the one-dimensional subspace spanned by k_l, and H_* which is the subspace (orthogonal to $\sum\oplus \{k_l\}$) satisfying $R_\nu f = 0$, $\nu = 0, 1, \ldots, m-1$, that is,

$$W_2^m = \{k_0\} \oplus \{k_1\} \oplus \cdots \oplus \{k_{m-1}\} \oplus H_*.$$

This construction can be found in e.g. Craven and Wahba(1979). Letting $\overset{d}{\otimes} W_2^m$ be the tensor product of W_2^m with itself d times, we have

$$\overset{d}{\otimes} W_2^m = \overset{d}{\otimes} [\{k_0\} \oplus \cdots \oplus \{k_{m-1}\} \oplus H_*]$$

and $\overset{d}{\otimes} W_2^m$ may be decomposed into the direct sum of $(m+1)^d$ fundamental subspaces, each of the form

$$[\ ] \otimes [\ ] \otimes \cdots \otimes [\ ] \quad (d \text{ boxes})$$

where each box ([ ]) is filled with either $\{k_l\}$ for some l, or H_*.

The subspace of $\otimes W_2^m$ for additive splines is the direct sum of all fundamental subspaces for which at most one of the boxes is filled with a symbol other than $\{k_0\}$. In the additive model $f(x_1, \cdots, x_d)$ is of the form

$$f(x_1, \ldots, x_d) = \mu + \sum_{\alpha=1}^{d} g_\alpha(x_\alpha)$$

where $g_\alpha \in \{k_1\}\oplus \cdots \oplus\{k_{m-1}\}\oplus H_*$ and the penalty term in (2.2) can be taken as

$$\sum_{\alpha=1}^{d} \theta_\alpha^{-1} \int_0^1 \left\{\frac{\partial^m g_\alpha}{\partial x_\alpha^m}\right\}^2 dx_\alpha.$$

Then we have the identifications: H_0 is the direct sum of the fundamental spaces with all k_0's in the boxes except at most a single k_l with l not equal to 0, and the H^β's are the fundamental spaces with all k_0's in the boxes except exactly one with H_*. The subspace for (all) two factor interactions is the direct sum of all fundamental subspaces for which exactly 2 boxes are filled with a symbol other than k_0, etc. For $m=1$, there is only one kind of 2 factor interaction fundamental subspace, it has 2 H_*'s in the boxes and $d-2$ k_0's, but for $m\geq 2$, there are subspaces with 2 H_*'s, with 1 H_* and one $k_l, l>0$, and with 2 elements of the form $k_l, l>0$, we shall call these pure, mixed, and parametric subspaces, respectively. Of course the possibilities multiply if one wishes to consider higher order interactions.

The form of the induced norms on the various subspaces can most easily be seen by an example. Suppose $d = 4$ and consider for example the subspace

$$[\{k_l\}] \otimes [H_*] \otimes [H_*] \otimes [\{k_r\}],$$

which we will assign the index $l^{**}r$. Then the square norm of the projection of f in $\overset{4}{\otimes} W_2^m$ onto this subspace is

$$\|P_{l^{**}r} f\|^2 =$$

$$\int_0^1\int_0^1 \left[\frac{\partial^{2m}}{\partial x_2^m \partial x_3^m} \int_0^1\int_0^1 R_{l(x_1)}R_{r(x_4)}f(x_1,x_2,x_3,x_4)dx_1 dx_4\right]^2 dx_2 dx_3,$$

where $R_{k(x_\alpha)}$ means R_k applied to what follows as a function of x_α. Using the fact that the reproducing kernel for $\{k_l\}$ is $k_l(x)k_l(x')$ and the rk for H_* is $Q(x\,;x')$ given by

$$Q(x\,;x') = k_m(x)k_m(x') - k_{2m}([x-x'])$$

where $[u]$ is the fractional part of u (see Craven and Wahba(1979)), it is easy to see that the r.k. for this subspace, call it

$$Q_{l^{**}r}(x_1,x_2,x_3,x_4;\ x'_1,x'_2,x'_3,x'_4) = Q_{l^{**}r}(x;x'),$$

is

$$Q_{l^{**}r}(x\,;x') =$$

$$k_l(x_1)k_l(x'_1)Q(x_2\,;x'_2)Q(x_3\,;x'_3)k_r(x_4)k_r(x'_4).$$

The rk for the direct sum of any number of fundamental subspaces is the sum of the rk's, since these fundamental subspaces are all orthogonal.

We now have a very flexible model building tool, by constructing models based on subspaces of interest. To discuss some of the possibilities in a simple way, let us first restrict ourselves to the case $m=1$, and consider only main effects and two factor interactions. Here H_0 is just the space of constants, there are d one factor subspaces, one for each variable, each of which is a fundamental space with all $\{k_0\}$'s in the boxes except one H_*, and $d\cdot(d-1)/2$ two factor spaces. Assigning each space its own θ_β (more precisely $\lambda\theta_\beta^{-1}$) and trying to estimate all these parameters, appears tricky even for small d, and, in fact we would like to eliminate interaction terms, and even main effect terms,

if they are not supported by the data. My students C. Gu and Z. Chen have been investigating various philosophical and numerical strategies for deciding to eliminate or add subspaces. The basic tool is the use of the algorithm in Gu et al. (June 1988), where we find that it is quite feasible to optimize $V(\lambda,\theta)$ for θ with one, two or possibly three components, with n of the order of hundreds. Therefore, to use the algorithm, we combine subspaces. Here, if we lump all additive spaces into one subspace (by taking the direct sum), and all interaction spaces into another, then there is only 1 component in θ. The rk Q_θ of (2.3) is constructed and λ and θ estimated by minimizing V of (2.4). If the estimated θ is sufficiently small, then the interaction spaces can be deleted. Various strategies for deciding what constitutes "sufficiently small" are being investigated. Other strategies consist of deleting individual interaction terms, examining main effects terms whose interactions have been deleted, etc. Proceeding to the $m=2$ case, possible strategies multiply quickly, however, preliminary fitting with $m=1$ can act as a screening tool (C. Gu, personal communication). We remark, at this point, that if one believes that the function one is trying to estimate is a sample function from the prior associated with the penalized least squares problem then the GML estimates of λ and θ are the minimizers of

$$M(\lambda,\theta) = \frac{z'(n\lambda J + \Sigma_\theta)^{-1} z}{[det(n\lambda J + \Sigma_\theta)^{-1}]^{\frac{1}{n}}}$$

and one has likelihood ratio tests for the presence or absence of various components. Testing that f is in H_0 even in the abstract case discussed at the start of this section is relatively straightforward because the test statistic $M(\hat{\lambda},\hat{\theta})/M(\lambda=\infty)$ is independent of where in H_0 f is, and (provided not too many free θ's are allowed), the distribution of the statistic under the null hypothesis can be obtained by Monte Carlo methods. See also Barry and Hartigan (1988), Wahba (March, 1987). The locally most powerful test, and a test based on GCV can be found in Cox et al. (1988). The relative behavior of these tests in the case where f is in H and $\|P_1 f\|$ is not equal to 0 (which guarantees that f is not a sample function) is not known. We note that looking at estimated main effects curves and two factor interaction surfaces is an interesting way to visualize functions of many variables, and we believe that these models have great potential for organizing large multivariate response data in a semiparametric way.

3. ACKNOWLEDGEMENTS This research supported in part by AFOSR under Grant AFOSR 87-0171, NSF under Grant ATM-840373, and NASA under Contract NAG5-316.

REFERENCES

Abromowitz, M., and Stegun, I. A. (1965), in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Washington, D. C.: U. S. Gov't Printing Office.

Barry, D. (1983) "Nonparametric Bayesian Regression." thesis, Yale University.

Barry, D. (1986), "Nonparametric Bayesian Regression," *Ann. Statist.*, 14, 934-953.

Barry, D., and Hartigan, J. (1988), "An Omnibus Test for Departures from Constant Mean, Manuscript".

Cox, D., Koh, E., Wahba, G., and Yandell, B. (1988), "Testing the (Parametric) Null Model Hypothesis in (Semiparametric) Partial and Generalized Spline Models," *Ann. Statist.*, 16, 113-119.

Friedman, J. H., Grosse, E., and Stuetzle, W. (1983), "Multidimensional Additive Spline Approxmation," *SIAM J. Sci.Stat. Comput.*, 4, 291-301.

Friedman, J., and Silverman, B. (1987) Flexible parsimonious smoothing and additive modeling Stanford Linear Accelerator, Stanford, CA SLAC-PUB-4390.

Friedman, J. H. (August 1988) Fitting functions to noisy data in high dimensions, Department of Statistics, Stanford University, manuscript.

Gu, C. (June 1988) RKPACK - a general purpose minipackage for spline modeling, Technical Report 832, Madison, WI: University of Wisconsin-Madison, Statistics Dept..

Gu, C., Bates, D. M., Chen, Z., and Wahba, G. (June 1988) The computation of GCV functions through Householder tridiagonalization with application to the fitting of interaction spline models, Technical Report 823, Madison, WI: University of Wisconsin-Madison Statistics Dept..

Hastie, T., and Tibshirani, R. (1987), "Generalized Additive Models: Some Applications," *J. Amer. Statist. Assoc.*, 82, 371-386.

Hollingsworth, A., and Lonnberg, P. (1986), "The Statistical Structure of Short Range Forecast Errors as Determined from Radiosonde Data. Part I: The Wind Field," *Tellus*, 38A, 111-136.

Kimeldorf, G., and Wahba, G. (1971), "Some Results on Tchebycheffian Spline Functions," *J. Math. Anal. Applic.*, 33, 82-95.

Lonnberg, P., and Hollingsworth, A. (1986), "The Statistical Structure of Short Range Forecast Errors as Determined from Radiosonde Data. Part II: The Covariance of Height and Wind Errors," *Tellus*, 38A, 137-161.

Stein, M.L. (1987), "Minimum Norm Quadratic Estimation of Spatial Variograms," *J. Am. Statist. Assoc.*, 82, 765-772.

Stein, M.L. (1988), "Asymptotically Efficient Prediction of a Random Field with with a Misspecified Covariance Function," *Ann. Statist.*, 16, 55-63.

Stone, C.J. (1985), "Additive Regression and Other Nomparametric Models," *Ann. Statist.*, 13, 689-705.

Wahba, G. (1975) "A Canonical Form for the Problem of Estimating Smooth Surfaces." University of Wisconsin-Madison, Statistics Department Technical Report #420.

Wahba, G. (1978), "Improper Priors, Spline Smoothing and the Problem of Guarding Against Model Errors in Regression," *J. Roy. Stat. Soc., Ser.B*, 40, 364-372.

Wahba, G., and Wendelberger, J. (1980), "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross-Validation," *Monthly Weather Review*, 108, 1122-1145.

Wahba, G. (1985), "A Comparison of GCV and GML for Choosing the Smoothing Parameter in the Generalized Spline Smoothing Problem," *Ann. Statist.*, 13, 1378-1402.

Wahba, G. (1986), "Partial and Interaction Splines for the Semiparametric Estimation of Functions of Several Variables," in *Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface*, ed. T.J. Boardman, Washington, D. C.: American Statistical Association, 75-80.

Wahba, G. (1988), "On the Dynamic Estimation of Relative Weights for Observation and Forecast in Numerical Weather Prediction, University of Wisconsin-Madison Statistics Dept. Technical Report 818.," in *Proceedings of the International Workshop on Remote Sensing Retrieval Methods*, ed. A. Deepak, Hampton, VA: Deepak Publishing Co..

Wahba, G. (March, 1987) Spline models in statistics, notes for the CBMS conference held at Ohio State University.

COMPUTING EMPIRICAL LIKELIHOODS

Art Owen, Stanford University

Abstract

The empirical distribution function of a sample is often presented as a nonparametric maximum likelihood estimate of the sampling distribution. The likelihood function it maximizes can be used to define a likelihood ratio function. This empirical likelihood ratio function has some of the properties of parametric likelihood ratio functions; in particular a nonparametric version of Wilks's (1938) theorem holds.

Like the bootstrap, empirical likelihood allows the statistician to substitute computer power for distributional assumptions. The methods differ in that the bootstrap uses Monte Carlo sampling while empirical likelihood performs a number of numerical optimizations.

This paper describes how the numerical optimization required by empirical likelihood may be performed. The focus is on confidence regions for multivariate means, with extensions to statistics that are smooth functions of means. For a multivariate mean, the optimization problem is convex and so there are optimization methods guaranteed to find the unique global optimum from any starting point.

One by-product is an algorithm for determining whether a point in euclidean space is within the convex hull of a given set of points.

Key Words and Phrases. Bootstrap, confidence set, convex duality, empirical likelihood, likelihood ratio test, nonparametric likelihood.

1. Introduction. Let $X_1, X_2, \ldots$ be independent random vectors in $\mathbb{R}^p$, for $p \geq 1$, with common distribution function F_0. The empirical distribution

$$F_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$$

is well known to be the nonparametric maximum likelihood estimate of F_0 based on $X_1, \ldots, X_n$. Here δ_x denotes a point mass at x. The likelihood function that F_n maximizes is

$$L(F) = \prod_{i=1}^{n} F\{X_i\}$$

where $F\{X_i\}$ is the probability of $\{X_i\}$ under F.

It is less well known that the empirical likelihood ratio function

$$R(F) = L(F)/L(F_n)$$

can be used to construct nonparametric confidence regions and tests. Let T be a statistical functional from a set of distributions on $\mathbb{R}^p$, taking values in $\mathbb{R}^q$, and consider sets of the form

$$S = \{T(F) \mid R(F) \geq c\}.$$

Under mild conditions on T, this S may be used as a confidence region for $T(F_0)$. The set of $T(F)$ where $F \ll F_n$ (that is, where F is absolutely continuous with respect to F_n) has likelihood $L(F) = 0$ if $F \neq F_n$.

The convex hull of the points X_i is the set $\{\int x\,dF \mid F \ll F_n\}$. To see this, let $\int x\,dF = \sum_{i=1}^{n} p_i X_i$ for some $p_i \geq 0$ with $\sum p_i = 1$. For small $\varepsilon > 0$ we have $R((1-\varepsilon)F_n + \varepsilon\delta_x) > 0$. By choosing x to range through $\mathbb{R}^p$ we can find the range of $\int x\,dF$ where $F \ll F_n$. The problem can be recast as one of finding probability distributions F that are concentrated on a specified set of points and have a specified mean. We restrict attention to distributions with support in the sample, that is, to distributions $F \ll F_n$. This is convenient because the statistician might not be willing to specify a bounded support for F, and because it reduces the construction of S to a finite dimensional problem. Owen (1987) proves:

Theorem 1. *Let $X, X_1, X_2, \cdots$ be i.i.d. random vectors in $\mathbb{R}^p$, with $E(X) = \mu_0$, and $\mathrm{var}(X) = \Sigma$ of rank $q > 0$. For positive $c < 1$ let $S_{c,n} = \{\int x\,dF \mid R(F) \geq c,\ F \ll F_n\}$. Then $S_{c,n}$ is a convex set and*

$$\lim_{n\to\infty} P(\mu_0 \in S_{c,n}) = P(\chi^2_{(q)} \leq -2\log c).$$

Moreover if $E(\|X\|^4) < \infty$ then

$$|P(\mu_0 \in S_{c,n}) - P(\chi^2_{(q)} \leq -2\log c)| = O(n^{-1/2}).$$

This is a nonparametric version of *Wilks's (1938) theorem*. The rate attained is also the same as Wilks finds. DiCiccio Hall and Romano (1988) show that when $E(\|X\|^6) < \infty$ the convergence in theorem 1 is at rate n^{-1}.

The computational problem that arises is the computation of the empirical profile likelihood ratio function

$$r(\mu) = \sup\{R(F) \mid \int x\,dF = \mu,\ F \ll F_n\} \tag{1.1}$$

for various candidates μ for $E(X)$. Tests of $E(X) = \mu$ are rejected if $r(\mu)$ is small and confidence regions are formed from the unrejected values of μ. It is convenient to plot $r(\mu)$ when $p \leq 2$.

Owen (1987) shows that for $\mu_0 = E(X)$

$$-2\log r(\mu_0) = T^2 + O_p(n^{-1/2})$$

where T^2 is Hotelling's statistic. This suggests referring to $(n-1)q/(n-q)F_{q,n-q}$ instead of the chisquare limit. The $n^{-1/2}$ term is a skewness term, that extends the confidence regions in directions of positive skewness.

The connections between empirical likelihood and other work, especially the bootstrap, is outlined in Owen (1987), from which most of this article is taken.

2. Computing for the Mean. We need to compute the function $r(\mu)$ given by (1.1) for various candidates μ for $E(X)$.

Owen (1987) shows that the problem may be reduced to maximizing $\prod np_i$, or equivalently

$$\sum \log p_i \tag{2.1}$$

subject to

$$p_i \geq 0,\quad \sum p_i = 1 \quad \text{and} \quad \sum p_i X_i = \mu.$$

This is clear when there are no ties in the data, but it holds even when there are ties. When μ is not in the convex hull of the data, the constraints cannot be satisfied. When μ is in the convex hull of the data, the problem is reduced by Lagrange multipliers to maximizing $\sum \log(1 + \lambda'(X_i - \mu))$ with respect to $\lambda \in \mathbb{R}^p$, which is

$$0 = \sum \frac{X_i - \mu}{1 + \lambda'(X_i - \mu)}. \tag{2.2}$$

The maximizing weights are given by

$$w_i = \frac{1}{n}\frac{1}{1+\lambda'(X_i-\mu)} \tag{2.3}$$

so that

$$r(\mu) = \prod (1+\lambda'(X_i-\mu))^{-1}.$$

For $p = 1$ it is easy to solve (3.2) with a safeguarded zero finding algorithm such as Brent's method (Press et al. 1986). Owen (1988) uses Brent's method to maximize empirical likelihood ratios for certain M-estimates. The bisection algorithm, which safeguarded zero finders use, does not exist for $p > 1$, so we reformulate the problem.

It is more convenient to consider finding a zero of $-g(\lambda)$. Inspection of $-g$ shows that it is the gradient with respect to λ of

$$f(\lambda) = -\sum \log(1+\lambda'(X_i-\mu)) \tag{2.4}$$

so a zero of g is a critical point of f.

We now argue that f is convex over a convex domain. Since we only need to consider λ for which all $w_i \le 1$, we may assume that

$$1+\lambda'(X_i-\mu) \ge 1/n > 0, \quad 1 \le i \le n \tag{2.5}$$

so λ may be confined to the intersection D of half spaces *defined by* (2.5). D *is convex. The Hessian of* f is

$$H(\lambda) = \sum \frac{(X_i-\mu)(X_i-\mu)'}{[1+\lambda'(X_i-\mu)]^2}$$

which is positive semidefinite on D *and hence* f *is convex.* When the sample variance of the X_i is of full rank, H is positive definite on D. We assume from now on that H is positive definite on D, and hence that f is strictly convex on D. It follows that the solution of (2.2) is the unique global minimum of f on D.

We now have the following dual problem: to maximize (2.1) over the unit simplex subject to p constraints is to minimize f over D without constraints. The first problem is in the $n-1$ independent variables of the simplex and the second is in the p components of λ, so that the unconstrained dual problem is generally of much smaller dimension than the original constrained problem.

Notice that $f(\lambda)$ is the log likelihood ratio function (2.1) with (2.3) substituted for w_i. Interestingly, this makes the dual problem one of minimum likelihood. For values of $\lambda \in D$ other than the solution of (2.2), the w_i in (2.3) need not sum to 1.

It is convenient to extend f from D to $\mathbb{R}^p$, while preserving the convexity. By (2.5) this may be done by replacing log in (2.4) by $\log^*$ defined by $\log^*(x) = \log(x)$ for $x > n^{-1}$ and $\log^*(x) = q(x)$ otherwise, where $q(x)$ is a quadratic function that matches the logarithm and its first two derivatives at n^{-1}.

There are a number of theorems that guarantee that the unique global minimizer of (2.4) can be found. Theorem 4.14 of Rheinboldt (1974, p. 51) covers three such algorithms that all change one component of λ at a time. No condition is placed on the starting point. Theorem 5.2 of Rheinboldt (1974 p. 62) guarantees superlinear convergence for the Davidon Fletcher Powell algorithm, provided that the starting point for λ is one for which the level set is compact. This holds for the problem at hand if we start at the origin.

It is unusual to have such strong theoretical support for a minimization problem. More commonly one can only get local convergence results that guarantee convergence to a relative minimum provided the starting point is sufficiently close to the solution. The problem at hand, the minimization of a convex function over a convex domain is known as convex programming. For a discussion of convex programming see Pshenichny and Danilin (1978, Chapter 3).

The convergence theorems describe the performance of the algorithms when computations are made with infinite precision, and infinite sequences of steps are carried out. In practice one has to contend with finite approximations on both issues. It has been the author's experience that the computations are most easily made for μ near $\bar{X}$ and that as μ approaches the convex hull of the data the computation becomes more difficult. Algorithms may therefore be compared on the basis of how small the log likelihood ratio must become before the algorithm encounters difficulty. A natural goal for computation is to be able to compute the log likelihood ratio down to values corresponding to confidence intervals with coverage well beyond that required in practice. For other values of μ the approximation $r(\mu) = 0$ is adequate. In the example of Section 3, the IMSL conjugate gradient routine ZXCGR applied to f extended via $\log^*$ allows computation of log likelihoods smaller than -50 which far exceeds the needs of any reasonable confidence regions for the mean.

When μ is outside of the convex hull of the data, there is no solution. In practice what happens is that the algorithm terminates at a large value of λ for which the slope of the logarithm is so small that the gradient is zero to the required precision. One can tell that this has happened because the w_i will no longer sum to 1. It may be more convenient to use this fact than to check whether a given point is within the convex *hull of the data, especially when the dimension of the* data is higher than 2.

3. Example. For an illustration we use some data from Larsen and Marx (1986, p. 440). Eleven male ducks, each a second generation cross between mallard and pintail, were examined. Their plumage was rated on a scale from 0 (completely mallardlike) to 20 (completely pintaillike) and their behavior was similarly rated on a scale from 0 (mallard) to 15 (pintail). Figure 1 shows these data, together with *nested empirical likelihood confidence contours for the mean.* The point with plumage=14 and behavior=11 is plotted with a circle of twice the area of the others, because it represents two ducks. The confidence contours are presented for nominal confidence levels: .50, .90, .95, .99, taken from 20/9 times the $F_{2,9}$ distribution. An asterisk marks the sample mean.

The dual problem from Section 2 was solved at each point in a 100 by 100 grid, using the IMSL conjugate gradient routine ZXCGR. Of the 10000 points approximately 35% of them were within the convex hull of the point cloud. The *computation started at the point nearest the sample mean,* and proceeded in a discrete counterclockwise spiral. For each point after the first, the starting value of λ in the optimization was taken from one of the neighboring points in the grid for which the empirical likelihood had already been computed. *The point chosen* was the one with the highest empirical likelihood. If all such points had an empirical likelihood ratio near zero the point was given an empirical likelihood ratio of zero. This saves considerable time outside the convex hull of the data. The computation took approximately 3 minutes on a VaxStation II (microvax) workstation.

In Figure 2, the same information is presented, with the contours now taken from a scaled $F_{2,9}$ distribution for Hotelling's T^2 statistic. These are parametric likelihood ra-

Figure 1: Empirical Likelihood Contours

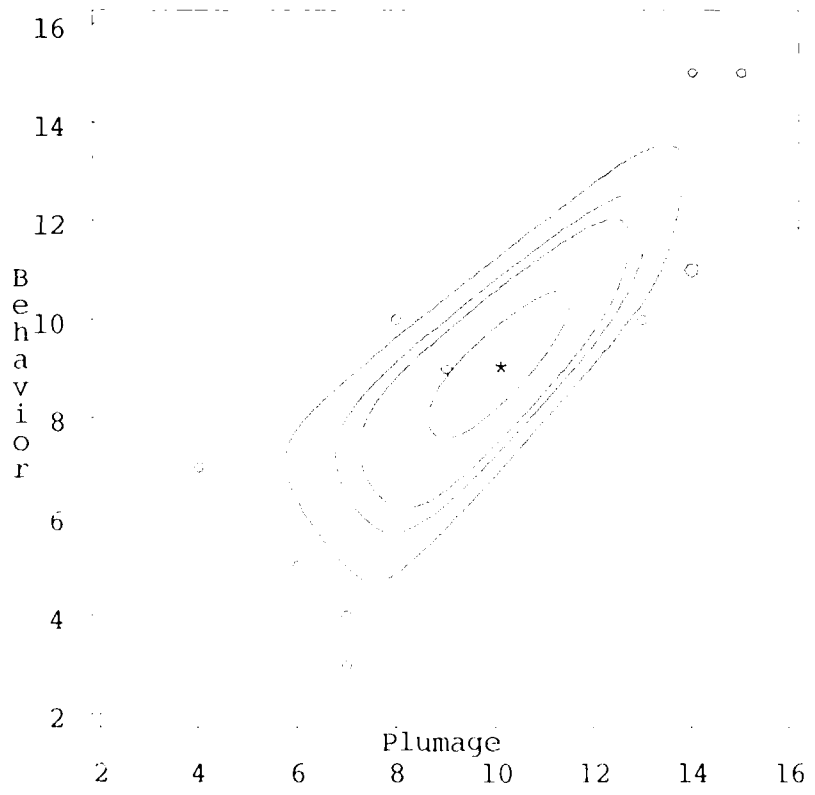
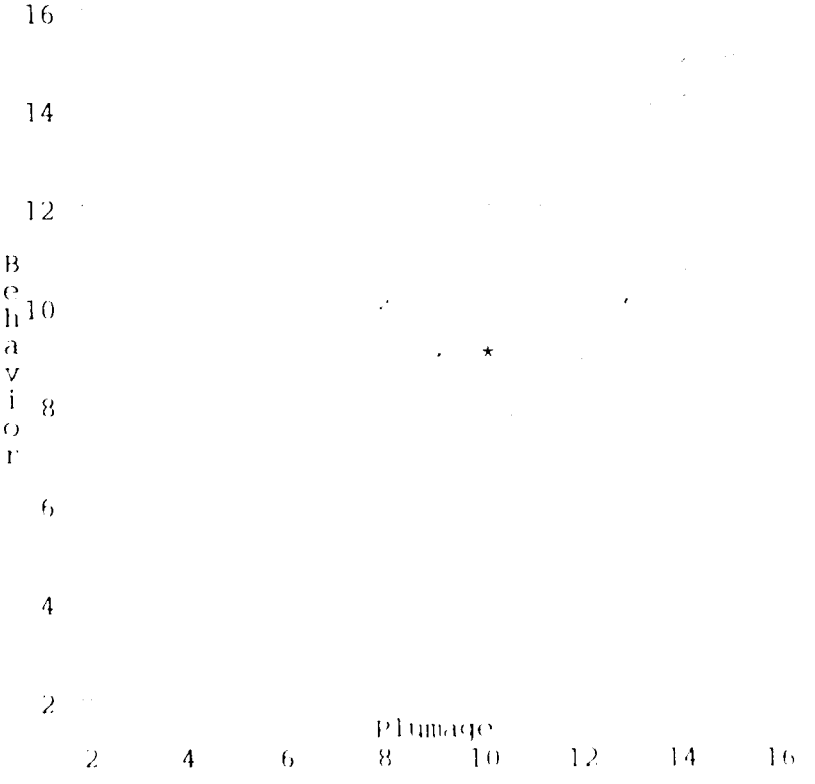


Figure 2: Normal Likelihood Contours



tio contours assuming a bivariate normal distribution with unknown mean and variance. The 50% region is nearly the same as for the empirical likelihood. More extreme regions remain elliptical while those of the empirical likelihood ratio method tend to the convex hull of the data.

4. Extensions to Other Statistics. Theorem 1 for means extends by delta method arguments to statistics that are smooth functions of means. See theorem 2 of Owen (1987). Examples include the variance of X which is a function of the mean of (X, X^2) and the correlation between X and Y which is a function of the mean of (X, Y, X^2, Y^2, XY). Similarly coefficients of skewness, partial correlation and regression can be treated this way. DiCiccio, Hall and Romano (1988) show that the coverage error is of order n^{-1} in this case under mild moment and derivative conditions. Extensions to M-estimates and to Frechet differentiable statistical functionals are made in Owen (1987). Joint confidence regions for p such statistics can be based on a chisquare limit with p degrees of freedom provided there are no linear dependencies among the statistics.

Consider the variance. The algorithm of Section 2 can be used to compute the empirical likelihood of the pair $(\mu, \mu^2 + \sigma^2)$ as a candidate for the mean of (X, X^2). For any fixed σ the resulting likelihood may be maximized over μ. Equivalently, we may compute the likelihood of (μ, σ^2) for the mean of $(X, (X-\mu)^2)$ and take as the likelihood of σ the maximum over μ. The latter prescription should be more stable numerically. Let

$$r(\mu, \sigma) = \sup \prod n w_i$$

subject to the constraints

$$w_i \geq 0, \quad \sum w_i = 1, \quad \sum w_i X_i = \mu, \quad \sum w_i (X_i - \mu)^2 = \sigma^2$$

and abuse notation by letting

$$r(\sigma) = \sup_{\mu} r(\mu, \sigma).$$

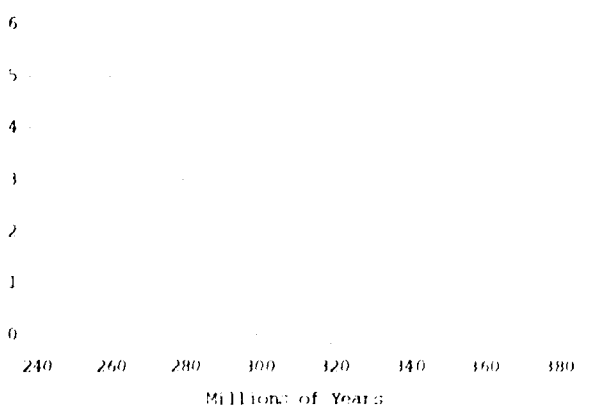
The analysis in Section 2 allows us to write $w_i = w_i(\lambda)$ where $\lambda \in \mathbb{R}^2$ is the Lagrange multiplier and now

$$\log r(\sigma) = \sup_{\mu} \inf_{\lambda} \log r'(\sigma, \mu, \lambda)$$

where

$$r'(\sigma, \mu, \lambda) = \prod \left(1 + \lambda_1 (X_i - \mu) + \lambda_2 ((X_i - \mu)^2 - \sigma^2)\right)^{-1}.$$

Figure 3: Potassium-Argon Dates



A nested optimization may be used to compute $r(\sigma)$. The inner level of the optimization minimizes the likelihood over λ with μ and σ held fixed. The outer level maximizes the resulting minimum over μ for fixed σ. In the outer level it is convenient to know the derivative of the minimum with respect to μ. This may be determined analytically,

$$\frac{d}{d\mu} \inf_{\lambda} \log r'(\sigma, \mu, \lambda) = n\lambda_1 \tag{4.1}$$

where λ_1 is the first component of the minimizing λ. A generic function optimizer may try impossible values of (μ, σ) before finding the optimum. This is especially likely for extremely large or small values of σ. It thus helps to extend the domain of the empirical likelihood as described in Section 2, through the function $\log^*$. The alternative is to design an optimization method more specific to empirical likelihood. When the likelihood is extended, the more general version of (4.1) is

$$\frac{d}{d\mu} \inf_{\lambda} \log r'(\sigma, \mu, \lambda) = n\lambda_1 \sum w_i(\lambda).$$

Larsen and Marx (1986, p. 332) give 19 estimated ages, in millions of years, of mineral samples collected in the Black Forest. The ages were estimated by Potassium-Argon dating. The variance of these measurements is of direct interest since it provides information on the precision of the dating method. A histogram of this data appears in Figure 3. The sample standard deviation is 27.1 million years, and a normal theory 95% confidence interval is 20.1 to 40 million years.

Figure 4: Likelihood Ratios

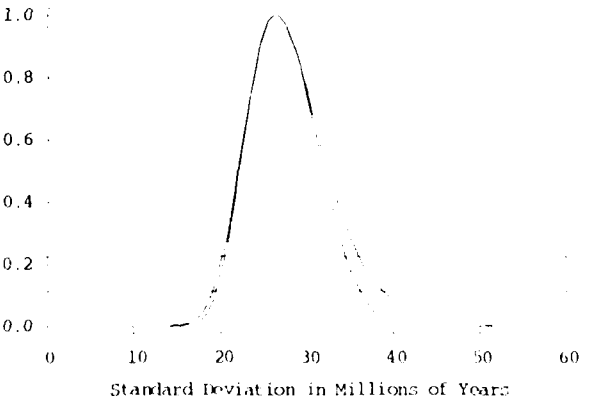
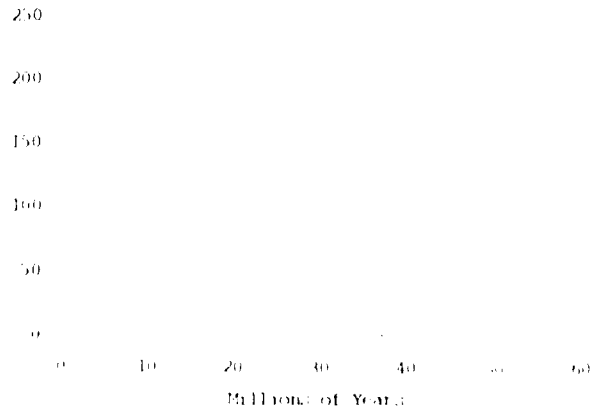


Figure 5: Bootstrap Standard Deviations



The empirical likelihood ratio was calculated for values of the variance corresponding to standard deviations in the range from 1.5 to 51.5 million years in steps of half a million years. Computations were made for an increasing sequence of standard deviations starting near the maximum likelihood estimate, and for a decreasing sequence starting there. This way the final values from each step could be used as starting values for the next. It took two minutes to make 102 likelihood evaluations on a microvax VaxStation II.

Figure 4 shows the empirical likelihood ratio function, together with the normal theory likelihood ratio function. The likelihood ratios are plotted against standard deviations in millions of years. The horizontal lines correspond to 90% and 95% empirical likelihood confidence intervals. Slightly different lines would be appropriate for the exact confidence regions based on a normal model. The empirical likelihood ratio curve has a shorter right tail and a very slightly longer left tail than the normal one. It is surprising how close the two curves are. The shorter right tail of the empirical curve seems natural given the apparent shortness of the tails in Figure 3. The sample kurtosis is 0.02 if one uses the normal maximum likelihood estimate of σ^2 as in Miller (1986, p. 272), and -0.29 if one uses the unbiased estimate of σ^2. Since the sample maximum is 344 and the minimum is 243, the largest possible standard deviation for a reweighted sample is 51.5. The algorithm found an empirical log likelihood of -52.9 for a standard deviation of 51. The smallest standard deviation for which a meaningful solution was obtained was 3.5 and the corresponding empirical log likelihood was -51.8. These correspond to putative $\chi^2_{(1)}$ values in excess of 100. It follows that for any confidence level of practical interest the empirical interval for the variance can be computed from this data. For standard deviations outside (3.5, 51) the modifications to the logarithm that make it possible to use generic optimizers lead to convergence to solutions for which the weights w_i sum to less than 1. It made the computations more stable to divide the ages by 100 before computing the intervals.

The normal theory curve is exact if the observations are normally distributed and has a large sample justification if the kurtosis of the measurements is 0. The empirical likelihood curve has a large sample justification provided that the kurtosis is finite. Figure 5 shows a histogram of 1000 bootstrap replications of the standard deviation. The histogram has a location and scale comparable to those of the likelihood ratio curves. The right tail of the bootstrap histogram looks more like the empirical likelihood ratio curve than the normal theory one.

A similar nested algorithm works for the correlation ρ. The inner level consists of finding the likelihood of

$$(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \rho\sigma_x\sigma_y)$$

as a mean for

$$(X, Y, (X - \mu_x)^2, (Y - \mu_y)^2, (X - \mu_x)(Y - \mu_y)).$$

The outer level consists of maximizing the result of the inner level over choices of $(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2)$. Using the dual problem, the inner optimization is over 5 variables and the outer is over 4 variables. The whole computation is done over a 1 dimensional grid of values for ρ. The 4 variables of the outer optimization must obey some constraints to be valid moments. Rather than check whether each trial point of the outer optimization is possible, it is easier to extend the inner function as described in Section 2. As before analytic derivatives are available for the outer optimization. Numerical performance is improved by centering and scaling both the X and Y variables.

Larsen and Marx (1986 page 456) give 15 pairs of observations relating the frequency with which crickets chirp to the temperature. The data are plotted in Figure 6. The frequency is measured in chirps per minute and the temperature is in degrees Fahrenheit. The sample correlation is 0.835. The empirical likelihood ratio function is plotted in Figure 7. Also shown is the normal theory profile likelihood ratio function. The empirical curve lies above the normal one. Figure 8 is a histogram of 1000 bootstrap replications of the correlation. The empirical likelihood ratio curve is very asymmetric, so it will yield inferences quite different from those based on an estimated standard deviation for ρ. The shape of the curve is similar to the bootstrap histogram.

Figure 6: Temperature vs. Chirps/Minute

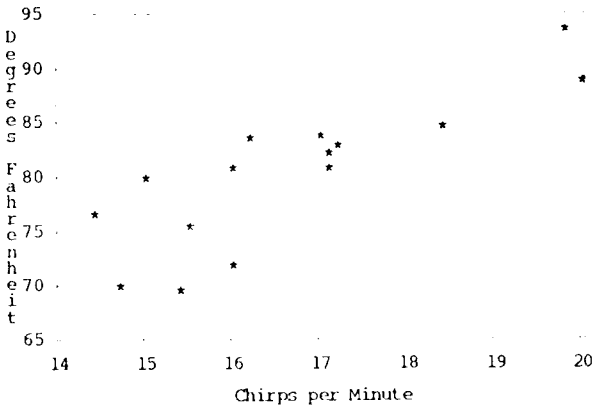


Figure 7: Likelihood Ratios

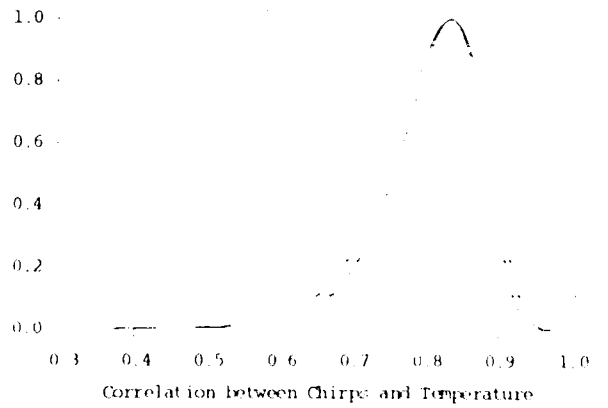


Figure 8: Bootstrap Correlations

7. Acknowledgements. The author would like to thank Nils Hjort, Michael Steele, Bradley Efron and Tom DiCiccio for helpful comments. This research was supported by National Science Foundation Grant DMS86-00235.

REFERENCES

DiCiccio, T.J., Hall, P.J. & Romano J.P. (1988). " Comparison of Parametric and Empirical Likelihood Functions". Technical Report No. 291, Dept. of Statistics, Stanford University.

Larsen, R.J. & Marx, M.L. (1986). *An Introduction to Mathematical Statistics and its Applications*. Prentice-Hall, Englewood Cliffs, New Jersey.

Miller, R.G. (1986). *Beyond ANOVA, Basics of Applied Statistics*. New York: J. Wiley & Sons.

Owen, A.B. (1987). "Empirical Likelihood Ratio Confidence Regions ". Technical Report No. 283, Dept. of Statistics, Stanford University.

Owen, A.B. (1988). Empirical Likelihood Ratio Confidence Intervals For a Single Functional. *Biometrika* **75** No. 2.

Pshenichny, B.N. & Danilin, Yu.M. (1978). *Numerical Methods in Extremal Problems*. Moscow: Mir Publishers.

Press, W.H., Flannery, B.P., Teukolsky, S.A. & Vetterling, W.T. (1986). *Numerical Recipes*. Cambridge: Cambridge University Press.

Rheinboldt, W.C. (1974). *Methods for Solving Systems of Nonlinear Equations*. Conf. Series in Appl. Math., No. 14. Philadelphia: SIAM.

Wilks, S.S. (1938). The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *Ann. Math. Statist.* **9**, 60 62.

COMPUTING EXTENDED MAXIMUM LIKELIHOOD ESTIMATES FOR LINEAR PARAMETER MODELS

Douglas B. Clarkson, IMSL, Inc. and Robert I. Jennrich, UCLA

Summary

Methods are given for computing *extended maximum likelihood* estimates in which one or more parameter estimates are infinite at the supremum of the likelihood. The results are given for a broad class of regression-like models based on independent observations with linearly related parameters including, in particular, the generalized linear models. The estimation consists of two steps: a linear programming step to identify the infinite components, and a more conventional function optimization step to optimize the remaining finite components. Provision is made for nuisance parameters. Two algorithms are presented and examples illustrating their use are given.

1. Introduction

When a few elements of a vector of estimates are infinite at the supremum of a likelihood, standard computational algorithms fail since convergence to infinity is not possible. Haberman (1974, appendix B) defines and gives an example of such estimates for frequency data, but he gives no computational algorithm. He calls the estimates obtained *extended maximum likelihood* estimates. Because these estimates contain infinite values, they do not exist in the usual sense and other authors (Silvapulle and Burridge, 1986, Anderson and Albert, 1984) have felt that detecting the presence of infinite estimates is sufficient. The algorithms they developed check for "existence" (i.e., finiteness) of the estimates, and leave the user to respond to the situation by adjusting the model, obtaining additional data, or halting the analysis. Following Haberman (1974), we feel that information such as the optimal log-likelihood and parameter estimates associated with each observation is useful and can be obtained by computing the extended maximum likelihood estimates. We give efficient computer algorithms for such computations for models involving linearly related parameters, including the "generalized linear models" of Nelder and Wedderburn (1972), linear models in survival analysis, and censored regression models.

The next section discusses extended maximum likelihood estimation. Sections 3 and 5 give theorems related to the computation of these estimates, Sections 4 and 6 give the computational algorithms, and Section 7 gives some examples.

2. Extended Maximum Likelihood Estimates

Let Θ be a subset of $\Re^n$ and for $\theta \in \Theta$ let $\ell(\theta)$ be a log-likelihood corresponding to some observed data. Let $\bar{\Re}$ denote the extended real line $[-\infty, \infty]$. We say that $\hat{\theta} \in \bar{\Re}^n$ is an *extended maximum likelihood estimate* if $\hat{\theta}$ is a limit point of Θ and for every sequence $\theta_m \in \Theta$ that converges to $\hat{\theta}$,

$$\lim_{m \to \infty} \ell(\theta_m) = \sup_{\theta \in \Theta} \ell(\theta).$$

This is equivalent to saying that $\ell(\theta)$ can be continuously extended so that its domain includes $\hat{\theta}$ and that $\hat{\theta}$ is a maximum likelihood estimate over the extended domain. Note that an ordinary maximum likelihood estimate is an extended maximum likelihood estimate.

Consider the binomial log-likelihood

$$\ell(\pi) = c + y \log \pi + (n - y) \log(1 - \pi), \qquad 0 < \pi < 1.$$

If $y = 0$, then $\hat{\pi} = 0$ is not in the domain $0 < \pi < 1$ of ℓ, but is an extended maximum likelihood estimate. Similarly, under the logistic parameterization

$$\pi = \frac{e^{\eta}}{1 + e^{\eta}}, \tag{1}$$

$\hat{\eta} = -\infty$ is an extended maximum likelihood estimate.

Consider the normal distribution $N(\eta, 1)$ and a censored observation $y > y_o$. The log-likelihood is

$$\ell(\eta) = c + \log \int_{y_o}^{\infty} e^{-\frac{1}{2}(y - \eta)^2} dy, \qquad -\infty < \eta < \infty.$$

Here $\hat{\eta} = \infty$ is an extended maximum likelihood estimate.

Most of our discussion will deal with *linear parameter models*. By this we mean models involving independent observations $y_1, \ldots, y_n$ with probability density or mass functions of the form

$$f(y_i | \eta_i), \quad \eta_i = \mathbf{x}_i \boldsymbol{\beta}. \tag{2}$$

Such models are called "models with a linear part" by Stirling (1984), but this terminology is by no means standard. By either name, these models include the generalized linear models of Nelder and Wedderburn (1972) and, as Stirling points out, many other models as well, including, for example, the censored normal example above.

The parameter vector $\boldsymbol{\beta}$ in (2) is common to all of the observations, and is related to the scalar parameter η_i for the i^{th} observation by the vector $\mathbf{x}_i$ of covariate or

design values. The η_i are clearly linearly related and may be displayed in vector form as

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta},$$

where $\mathbf{X}$ contains the $\mathbf{x}_i$ as rows. Let $\mathcal{M}$ denote the column space of $\mathbf{X}$. Then $\boldsymbol{\eta}$ ranges over $\mathcal{M}$, which, for the purpose of our development, is an arbitrary subspace of $\Re^n$.

3. Results for Linear Parameter Models

Linear parameter models have likelihoods whose logarithms have the additive form

$$\ell(\boldsymbol{\eta}) = \sum_{i=1}^{n} \ell_i(\eta_i), \qquad \boldsymbol{\eta} = (\eta_i) \in \mathcal{M}, \tag{3}$$

where $\mathcal{M}$ is a subspace of $\Re^n$ and the functions ℓ_i in the sum are defined on the real line $\Re$.

Assuming it exists, let

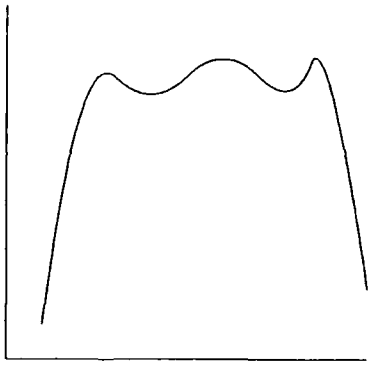
$$\ell_i(\infty) = \lim_{t \to \infty} \ell_i(t),$$

and define $\ell_i(-\infty)$ similarly. We will use the following additional assumptions about the functions ℓ_i:

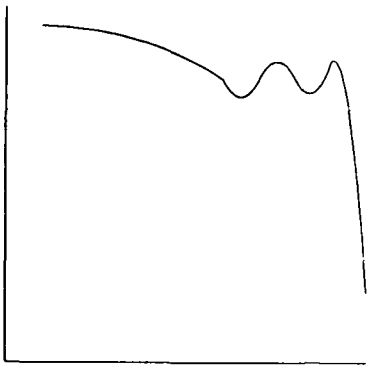
Assumptions: *For each* $i = 1, \ldots, n$

1. ℓ_i *is continuous and bounded on* $\Re$,
2. $\ell_i(\infty) = -\infty$,
3. $\ell_i(-\infty) = \begin{cases} -\infty, & \text{or} \\ \sup \ell_i. \end{cases}$

Under these assumptions the ℓ_i have one of the two forms displayed in Figure 1.



a. doubly descending



b. singly descending

Figure 1. Assumed Forms for the Likelihood Terms

We say those ℓ_i of form (a) are *doubly descending* and those of form (b) are *singly descending*. The choice of right descension for singly descending ℓ_i is arbitrary; a left descending ℓ_i can be made right descending by reparameterizing.

In the case of the binomial example with the logistic parameterization, there is only one ℓ_i and it is right descending when $y = 0$, left descending when $y = n$, and doubly descending otherwise. The censored normal ℓ_i is left descending. It, and the binomial ℓ_i when $y = n$, would have to be reparameterized to satisfy assumptions (2) and (3).

Given assumption (3), we make the following:

Definition 1: *Choose* $\boldsymbol{\eta}^* \in \mathcal{M}$ *so* $\boldsymbol{\eta}^* \leq 0$, $\eta_i^* = 0$ *whenever* $\ell_i(-\infty) = -\infty$, *and* $\boldsymbol{\eta}^*$ *has a maximum number of negative components. Let* $\mathcal{D} = \{i : \eta_i^* = 0\}$.

By considering convex combinations, it is easy to see that $\boldsymbol{\eta}^*$ exists and $\mathcal{D}$ is uniquely defined. The following theorem will be used to identify the finite part of an extended maximum likelihood estimate for $\boldsymbol{\eta}$ in (3).

Theorem 1: *Under assumptions (1), (2), and (3), if* $\mathcal{D}$ *is not empty,*

$$\ell^*(\boldsymbol{\eta}) = \sum_{i \in \mathcal{D}} \ell_i(\eta_i), \quad \boldsymbol{\eta} \in \mathcal{M} \tag{4}$$

has a maximum.

Proof: Assume ℓ^* does not has a maximum in $\mathcal{M}$. Choose a sequence $\boldsymbol{\eta}_m \in \mathcal{M}$ so that

$$\ell^*(\boldsymbol{\eta}_m) \to \sup \ell^*. \tag{5}$$

Choose a subsequence $\boldsymbol{\eta}'_m$ of $\boldsymbol{\eta}_m$ so that

$$\operatorname{dir} \boldsymbol{\eta}'_m \to \mathbf{d}.$$

Let $\boldsymbol{\eta}'_m = \rho'_m \mathbf{d}'_m$, where ρ'_m and $\mathbf{d}'_m$ are the length and direction of $\boldsymbol{\eta}'_m$. Since ℓ^* has no maximum, $\rho'_m \to \infty$

If $d_i > 0$ for some $i \in \mathcal{D}$, $\ell_i(\eta'_{im}) \to -\infty$ by assumption (2). Since each ℓ_i is bounded above by assumption (1), $\ell^*(\boldsymbol{\eta}'_m) \to -\infty$. This contradicts (5), and hence $d_i \leq 0$ for all $i \in \mathcal{D}$. Assume $d_i < 0$ and $\ell_i(-\infty) = -\infty$ for some $i \in \mathcal{D}$. Again, $\ell^*(\boldsymbol{\eta}'_m) \to -\infty$ and this contradiction implies $d_i = 0$ whenever $\ell_i(-\infty) = -\infty$ and $i \in \mathcal{D}$. Since by the definition of $\mathcal{D}$, $\ell_i(-\infty) = -\infty$ implies $i \in \mathcal{D}$, $d_i = 0$ whenever $\ell_i(-\infty) = -\infty$.

Assume $d_i = 0$ for all $i \in \mathcal{D}$. Then $\ell^*(0) = \ell^*(\boldsymbol{\eta}'_m)$ and by (5), $\ell^*(0) = \sup_{\boldsymbol{\eta} \in \mathcal{M}} \ell^*(\boldsymbol{\eta})$. This contradicts the assumption that ℓ^* has no maximum on $\mathcal{M}$. Thus $d_i < 0$ for at least one $i \in \mathcal{D}$.

Let $\boldsymbol{\eta}^*$ be any vector that defines $\mathcal{D}$ in Definition 1, and let

$$\mathbf{d} = \mathbf{d} + \alpha \boldsymbol{\eta}^*.$$

Then for α sufficiently large, $\mathbf{d} \leq 0$, and $d_i = 0$ whenever $\ell_i(-\infty) = -\infty$. Note, however, that $\mathbf{d}$ has at least one more negative component than $\boldsymbol{\eta}^*$. This contradicts the definition of $\boldsymbol{\eta}^*$ and implies that ℓ^* has a maximum on $\mathcal{M}$.

The following theorem will tell how to construct extended maximum likelihood estimates.

Theorem 2: *Under assumptions (1), (2), and (3),*

$$\sup \ell = \sup_{\boldsymbol{\eta} \in \mathcal{M}} \sum_{i \in \mathcal{D}} \ell_i(\eta_i) + \sum_{i \in \mathcal{D}^C} \sup \ell_i. \tag{6}$$

Proof: Let $\hat{\boldsymbol{\eta}} \in \mathcal{M}$ maximize ℓ^* as defined in Theorem 1 and let $\boldsymbol{\eta}^*$ be as defined in Definition 1. Then for any $\alpha > 0$,

$$\sup \ell \geq \ell(\hat{\boldsymbol{\eta}} + \alpha \boldsymbol{\eta}^*) = \sup \ell^* + \sum_{i \in \mathcal{D}^C} \ell_i(\hat{\eta}_i + \alpha \eta_i^*).$$

For $i \in \mathcal{D}^C$, as $\alpha \to \infty$, $\ell_i(\hat{\eta}_i + \alpha \eta_i^*) \to \ell_i(-\infty)$. Since $\ell_i(-\infty) = \sup \ell_i$ for $i \in \mathcal{D}^C$,

$$\sup \ell \geq \sup \ell^* + \sum_{i \in \mathcal{D}^C} \sup \ell_i.$$

The opposite inequality is easy.

To construct an extended maximum likelihood estimate under assumptions (1), (2), and (3), find $\mathcal{D}$ as given in Definition 1. Then find a maximizer $\hat{\boldsymbol{\eta}} \in \mathcal{M}$ of ℓ^* as defined by (4). This exists by Theorem 1. Since under these same assumptions $\sup \ell_i = \ell_i(-\infty)$ for each $i \in \mathcal{D}^C$, it follows from Theorem 2 that

$$\hat{\eta}_i = \begin{cases} \hat{\eta}_i & i \in \mathcal{D} \\ -\infty & i \in \mathcal{D}^C \end{cases}$$

is an extended maximum likelihood estimate. Thus, in theory at least, one simply needs to find $\mathcal{D}$ and maximize ℓ^*.

The next section will show how to construct $\mathcal{D}$ using linear programming. This will be combined with an optimization program to give the first of the algorithms of section 5. The second algorithm uses an initial run of the optimization program and the following theorem to help find $\mathcal{D}$.

Theorem 3: *If under assumptions (1), (2), and (3) each singly descending ℓ_i has the property that*

$$\ell_i(t) < \ell_i(-\infty) \tag{7}$$

for all t and if

$$\ell_{\mathcal{S}}(\boldsymbol{\eta}) = \sum_{i \in \mathcal{S}} \ell_i(\eta_i)$$

has a maximum at $\hat{\boldsymbol{\eta}} \in \mathcal{M}$, then $\mathcal{S} \subseteq \mathcal{D}$.

Proof: Assume $\mathcal{S}$ is not a subset of $\mathcal{D}$. Using $\boldsymbol{\eta}^*$ from Definition 1, consider

$$\ell_{\mathcal{S}}(\hat{\boldsymbol{\eta}} + \rho \boldsymbol{\eta}^*) = \sum_{i \in \mathcal{S} \cap \mathcal{D}} \ell_i(\hat{\eta}_i) + \sum_{i \in \mathcal{S} \cap \mathcal{D}^C} \ell_i(\hat{\eta}_i + \rho \eta_i^*).$$

For each ℓ_i in the second sum

$$\ell_i(\hat{\eta}_i + \rho \eta_i^*) \to \ell_i(-\infty)$$

as $\rho \to \infty$. Since each of these ℓ_i is singly descending by Definition 1, it follows from (7) that for some ρ sufficiently large,

$$\ell_i(\hat{\eta}_i + \rho \eta_i^*) > \ell_i(\hat{\eta}_i)$$

for each such ℓ_i, and hence that

$$\ell_{\mathcal{S}}(\hat{\boldsymbol{\eta}} + \rho \boldsymbol{\eta}^*) > \ell_{\mathcal{S}}(\hat{\boldsymbol{\eta}}).$$

This contradicts the assumption that $\hat{\boldsymbol{\eta}}$ maximizes $\ell_{\mathcal{S}}$.

4. Finding $\mathcal{D}$ by Linear Programming

Recall the parameterization $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ introduced in Section 2. Let $\mathbf{X}_1$ denote the submatrix of $\mathbf{X}$ containing the rows that correspond to the doubly descending ℓ_i (and possibly additional rows with indices known to be in $\mathcal{D}$) and let $\mathbf{X}_2$ denote the submatrix obtained from the remaining rows. From Definition 1 we seek a $\boldsymbol{\beta}$ such that

$$\begin{aligned} \mathbf{X}_1\boldsymbol{\beta} &= 0, \\ \mathbf{X}_2\boldsymbol{\beta} &\leq 0, \end{aligned} \tag{8}$$

and $\mathbf{X}_2\boldsymbol{\beta}$ has as many negative components as possible. If the columns of $\mathbf{X}_1$ are linearly independent, $\boldsymbol{\beta} = 0$ is the only solution to (8) and $\mathbf{X}_2\boldsymbol{\beta}$ can have no negative components. Then $\mathcal{D}$ is the complete set of integers from 1 to n. Otherwise, using appropriate row transformations,

(8) can be put in the equivalent form

$$\begin{aligned} \mathbf{X}_{11}\boldsymbol{\beta}_1 + \mathbf{X}_{12}\boldsymbol{\beta}_2 &= 0, \\ \tilde{\mathbf{X}}_{22}\boldsymbol{\beta}_2 &\leq 0, \end{aligned} \tag{9}$$

where $\mathbf{X}_{11}$ is a submatrix of $\mathbf{X}_1$ with linearly independent columns spanning the space of $\mathbf{X}_1$. The problem of finding $\mathcal{D}$ is reduced to finding a $\boldsymbol{\beta}_2$ such that $\tilde{\mathbf{X}}_{22}\boldsymbol{\beta}_2 \leq 0$ and $\tilde{\mathbf{X}}_{22}\boldsymbol{\beta}_2$ has as many negative components as possible. This is equivalent to finding $\boldsymbol{\beta}_2$ and $\boldsymbol{\delta}$ so that

$$\begin{aligned} \tilde{\mathbf{X}}_{22}\boldsymbol{\beta}_2 + \boldsymbol{\delta} &= 0, \\ \boldsymbol{\delta} &\geq 0, \end{aligned} \tag{10}$$

and $\boldsymbol{\delta}$ has as many positive components as possible. Using row transformations to eliminate the "free variables" $\boldsymbol{\beta}_2$ as described by Luenberger (1984, page 13), (10) can be written in the equivalent form

$$\begin{aligned} A\boldsymbol{\delta}_1 + \boldsymbol{\delta}_2 &= 0, \\ \boldsymbol{\delta}_1, \boldsymbol{\delta}_2 &\geq 0, \end{aligned} \tag{11}$$

where $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ are subvectors of $\boldsymbol{\delta}$.

To find a $\boldsymbol{\delta}$ with as many positive components as possible, apply the simplex algorithm to maximize the function $f(\boldsymbol{\delta}) = \sum_i c_i\delta_i$ under the constraint (11), where initially each $c_i = 1$. Iterate until convergence or until all elements in the column about to enter the basis are nonpositive. At this point there is a feasible solution to (11) that has positive values for the variable about to enter and for all basic variables corresponding to negative values in the column about to enter. Rather than performing the pivot, set the coefficients c_i for all of these variables to zero. The reduced coefficient for the variable that was about to enter will now be zero. Continue with the simplex algorithm in this modified manner until it converges. Coefficients $c_i = 1$ at convergence identify elements in $\mathcal{D}$. More specifically, $\mathcal{D}$ is identified by the indices of the rows of $\mathbf{X}_1$ and the indices of the rows of $\mathbf{X}_2$ corresponding to $c_i = 1$ at convergence.

5. Nuisance parameters

For applications it is useful to deal with models that are a little more general than the linear parameter models of the previous section. Consider a log likelihood of the form

$$\ell(\boldsymbol{\eta}, \boldsymbol{\phi}) = \sum_{i=1}^{n} \ell_i(\eta_i, \boldsymbol{\phi}), \tag{12}$$

where $\boldsymbol{\eta} \in \mathcal{M}$ and $\boldsymbol{\phi} \in \Phi$ is a set of nuisance parameters.

If for each $\boldsymbol{\phi}$, the functions $\ell_i(t, \boldsymbol{\phi})$ of t satisfy assumptions (1), (2), and (3), and if the set of singly descending $\ell_i(t, \boldsymbol{\phi})$ remains the same for all values of $\boldsymbol{\phi}$, then by Theorem 2, $\ell(\boldsymbol{\eta}, \boldsymbol{\phi})$ and

$$\sum_{i \in \mathcal{D}} \ell_i(\eta_i, \boldsymbol{\phi}) + \sum_{i \in \mathcal{D}^c} \ell_i(-\infty, \boldsymbol{\phi}) \tag{13}$$

have the same supremum. If $(\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\phi}})$ maximizes (13) and

$$\hat{\eta}_i = \begin{cases} \tilde{\eta}_i & i \in \mathcal{D} \\ -\infty & i \in \mathcal{D}^c \end{cases}$$

then $(\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\phi}})$ is an extended maximum likelihood estimate.

6. Two Algorithms

Let $\mathcal{T}$ denote the set of indices i for which ℓ_i is doubly descending. The simplest algorithm consists of two steps. The first is to solve the linear programming problem in Section 4 with $\mathbf{X}_1$ containing the rows of $\mathbf{X}$ with indices in $\mathcal{T}$. This gives $\mathcal{D}$. A general optimization algorithm such as Fisher scoring or Newton-Raphson is then applied to maximize (13).

An alternative approach that simplifies the linear programming step, and avoids it entirely when an extended estimate is not required, proceeds as follows: Apply the optimizing algorithm to the complete $\ell(\boldsymbol{\eta}, \boldsymbol{\phi})$ as given by (12). If during the iteration a component η_i of $\boldsymbol{\eta}$ appears to be too negative and if $i \notin \mathcal{T}$, eliminate the term $\ell_i(\eta_i, \boldsymbol{\phi})$ from (12). Continue in this manner until no further terms are eliminated. Let $\mathcal{S}$ denote the indices of the terms which have not been eliminated. If the algorithm converges to a maximum of

$$\ell_{\mathcal{S}}(\boldsymbol{\eta}, \boldsymbol{\phi}) = \sum_{i \in \mathcal{S}} \ell_i(\eta_i, \boldsymbol{\phi}), \tag{14}$$

where $\boldsymbol{\eta} \in \mathcal{M}$ and $\boldsymbol{\phi} \in \Phi$, then by theorem 3, $\mathcal{S} \subseteq \mathcal{D}$ and the linear programming problem in Section 4 can begin with $\mathbf{X}_1$ corresponding to the rows of $\mathbf{X}$ with indices in $\mathcal{S}$. From this point the algorithm proceeds as in the first algorithm with $\mathcal{T}$ replaced by $\mathcal{S}$.

In practice one may or may not know one's algorithm produces a global optimum of (14). If, for example, one is only sure it produces a local maximum or perhaps a stationary point, then a formal appeal to Theorem 3 is not possible, but the second algorithm often seems to work nevertheless and may be usefully viewed as a heuristic algorithm.

There are often very practical methods for determining when an η_i is too negative. For example, in the logistic model, and in many other models, η_i is too negative when $\ell_i(\eta_i)$ is very close to 0.

7. Examples

It is easy to show that the assumptions of sections 3 and 5 hold for interval data in linear parameter models based upon binomial, negative binomial, logarithmic, log normal, exponential, Weibull, and extreme value distributions. The assumptions do not hold for the Cox

proportional hazards model, which can also suffer from the problems of infinite estimates. The algorithms may still yield extended maximum likelihood estimates, however. Further research is needed.

FORTRAN subroutines for the algorithms have been implemented and will eventually be available in the IMSL (1987) libraries. To illustrate the advantages of each of the algorithms, consider the data in Table 1. A two-way additive factorial logistic model with standard restrictions was fit to the data.

Table 1: *Logistic Regression, Binomial Data*

Obs.	Cell	η	y	n
1	(1,1)	$\mu+\alpha+\beta$	5	5
2	(1,2)	$\mu+\alpha-\beta$	3	5
3	(2,1)	$\mu-\alpha+\beta$	1	5
4	(2,2)	$\mu-\alpha-\beta$	0	5

With algorithm 1, observations 2 and 3 were identified as members of the set $\mathcal{D}$ in the linear programming step, and a quasi-Newton algorithm then required 3 iterations to converge to the optimal likelihood based upon these two observations. For algorithm 2 the quasi-Newton routine required 8 iterations and yielded observations 2 and 3 as members of the set $\mathcal{S}$. The linear programming algorithm then verified that observations 2 and 3 were the only elements in $\mathcal{D}$. The estimated coefficients μ, α, and β *were different in the two algorithms, but this is to be* expected since the estimated coefficients are not unique when infinite estimates occur. Also as expected, the predicted $\hat{\eta}_i$ and probabilities in each cell were identical, as were the optimal log-likelihoods.

A second example illustrating why algorithm 2 might be preferred uses the same data as in Table 1, but tabulated as Bernouli trials. This form for the data is given in Table 2. Under the column "Freq." is the number of observations for which the outcome applies. Thus there were 3 trials in which a "1" (success) was observed in cell (1,2), while 2 trials in this cell were "0" (failures).

Table 2: *Logistic Regression, Bernouli Data*

Obs.	Cell	η	Freq.	y
1	(1,1)	$\mu+\alpha+\beta$	5	1
2	(1,2)	$\mu+\alpha-\beta$	3	1
3	(1,2)	$\mu+\alpha-\beta$	2	0
4	(2,1)	$\mu-\alpha+\beta$	1	1
5	(2,2)	$\mu-\alpha+\beta$	4	0
6	(2,2)	$\mu-\alpha-\beta$	5	0

In algorithm 1 no observations were initially identified as members of the set $\mathcal{T}$, and the linear programming using all observations, selected observations 1 and 6 as observations with infinite η_i. The optimization routines required 3 iterations to converge. For algorithm 2 the optimization routine required 7 iterations and identified observations 1 and 6 as observations with potentially infinite η_i. The linear programming verified that the η_i for these observations were infinite. Because algorithm 1 must first solve a linear programming problem based upon all of the data, while algorithm 2 converged directly to the final solution, which was then verified by a smaller linear programming problem, algorithm 2 would probably be preferred. Obviously, this becomes more important as the size of the data set increases.

8. Conclusions

Some algorithms for computing extended maximum likelihood estimates, theorems justifying their use, and examples illustrating the performance of the algorithms have been presented. The theorems have been given for linear parameter models, but the algorithms may find application in a more general context. Of the two algorithms presented, algorithm 1 requires fewer assumptions and thus may be more robust. Algorithm 2, however, is often more convenient computationally, and may thus be preferred where applicable. Because infinite estimates are probably not too common, it would seem reasonable to apply the algorithms discussed here only after an algorithm for finite estimates had failed.

References

Albert, A. & Anderson J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*. **71**, 1 10.

Haberman, S. J. (1974). *The Analysis of Frequency Data*. Chicago: The University of Chicago Press.

Luenberger, D. G. (1984). *Linear and Nonlinear Programming, Second Edition*. Reading, Massachusetts: Addison-Wesley Publishing Company.

McCullagh, P. & Nelder J. A. (1983). *Generalized Linear Models*. London: Chapman and Hall.

Nelder, J. A., & Wedderburn R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*. **135**, 370 384.

Silvapulle, M. J. & Burridge J. (1986). Existence of maximum likelihood estimates in regression models for grouped and ungrouped data. *Journal of the Royal Statistics Society, Series B*. **48**, 100 106.

Stirling, D. W. (1984). Iteratively reweighted least squares for models with a linear part, *Applied Statistics*, **33**, 7 17.

Wedderburn, R. W. M. (1976). On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. *Biometrika*. **63**, 27 32.

SIMULTANEOUS CONFIDENCE INTERVALS IN THE GENERAL LINEAR MODEL

Jason C. Hsu, The Ohio State University

Abstract

Consider the general linear model (GLM) $\underline{Y} = \mathbf{X}\underline{\beta} + \underline{e}$. Suppose $\beta_1, \ldots, \beta_k$ ($k \le p$) are of interest; $\beta_1, \ldots, \beta_k$ may be treatment contrasts in an ANOVA setting, or regression coefficients in a response surface setting. Computing the coverage probability of simultaneous confidence intervals for $\beta_1, \cdots, \beta_k$ by iterated (k+1)-dimensional integration is impractical for all but the smallest data sets. We proposes to approximate the probability as a mixture of products of univariate normal probabilities so that the number of functional evaluations becomes *linear* in k. The performance of this approximation is demonstrated in a variety of settings.

1. Simultaneous Statistical Inference and the von Neumann Bottleneck

Consider the general linear model (GLM) $\underline{Y} = \mathbf{X}\underline{\beta} + \underline{e}$, where $\underline{Y}_{N\times 1}$ is the vector of observations, $\mathbf{X}_{N\times p}$ is a known design matrix, $\underline{\beta} = (\beta_0, \ldots, \beta_p)'$ is the vector of parameters, and $\underline{e}_{N\times 1}$ is a vector of *iid* Normal$(0,\sigma^2)$ errors with σ^2 unknown. Suppose $\beta_1, \ldots, \beta_k$ ($k \le p$) are of interest and are estimable; $\beta_1, \ldots, \beta_k$ may be treatment contrasts in an ANOVA setting, or regression coefficients in a response surface setting. Let $\hat{\underline{\beta}} = (\hat{\beta}_1, \cdots, \hat{\beta}_k)'$ denote the BLUE (best linear unbiased estimator) of $(\beta_1, \cdots, \beta_k)'$. Then $(\hat{\beta}_1, \cdots, \hat{\beta}_k)'$ is multivariate normal with mean $(\beta_1, \cdots, \beta_k)'$ and variance-covariance $\sigma^2\mathbf{V}$ where $\mathbf{V}$ is the k ×k sub-matrix of the generalized inverse of $\mathbf{X}'\mathbf{X}$ corresponding to $\beta_1, \cdots, \beta_k$. Assume $\mathbf{V}$ is non-singular and let s^2 = MSE denote the usual estimator of σ^2, $\nu s^2/\sigma^2$ has a χ^2 distribution with ν = N–p degrees of freedom.

To give two-sided simultaneous confidence intervals for $\beta_1, \cdots, \beta_k$ with exact coverage probability $1-\alpha$

$$P\{\hat{\beta}_i - |q|s\sqrt{v_{ii}} \le \beta_i \le \hat{\beta}_i + |q|s\sqrt{v_{ii}} \text{ for } i = 1, \cdots, k\} = 1-\alpha$$

we need the quantile $|q|$ such that

$$P\{\max_{i=1,\cdots,k} (|\hat{\beta}_i - \beta_i| / s\sqrt{v_{ii}}) \le |q|\} = 1-\alpha. \quad (1)$$

To solve for the quantile $|q|$, the probability $P\{\max_{i=1,\cdots,k} (|\hat{\beta}_i - \beta_i| / s\sqrt{v_{ii}}) \le |q|\}$ has to be computed for candidates $|q|$, which involves (k+1)-dimensional integration *if one naively integrates over s and* $\hat{\beta}_k, \cdots, \hat{\beta}_1$ *in turn*. If m-point univariate Gaussian quadrature is performed iteratively, then roughly $2m^{k+1}$ evaluations of the univariate normal distribution function Φ or equivalent (e.g. Schervish 1984) is required. Thus for all but the smallest k, the von Neumann bottleneck prevents the computation of this iterated integral from being practical for interactive statistical data analysis.

2. Existing Methods

Traditionally, this "curse of dimensionality" is sidestepped by replacing the multivariate normal computation by computations involving individual $|\hat{\beta}_i - \beta_i|$ (using the Bonferroni inequality or Sidak's inequality) or computations involving pairs of $|\hat{\beta}_i - \beta_i|$ and $|\hat{\beta}_j - \beta_j|$ (using the Hunter-Worsley inequality). As these procedures are based on conservative probablistic inequalities, the resulting simultaneous confidence intervals can be much wider than necessary. The projection method of Scheffé-Working-Hotelling avoids integration altogether by obtaining a $100(1-\alpha)\%$ confidence ellipsoid for $\underline{\beta}$ based on the F distribution, then projecting the ellipsoid onto the $\beta_1, \cdots, \beta_k$ axes. These projected confidence intervals for $\beta_1, \cdots, \beta_k$ tend to be even wider than the probablistic inequality confidence intervals. (see Fuchs and Sampson, 1987, and examples below)

Though not stated in *SAS User's Guide: Statistics* (1985, p. 448), the MEANS option of PROC GLM in SAS® for multiple comparisons ignores any covariates $\beta_{k+1}, \cdots, \beta_p$ in the user's model when estimating $\beta_1, \cdots, \beta_k$ (but not in estimating MSE). Clearly, conclusions reached without taking significant covariates into account can be totally misleading. (see Example 6.1 below) Unsuspecting users have analyzed and published scientific findings based on this option of PROC GLM stating incorrectly that covariates $\beta_{k+1}, \cdots, \beta_p$ have been adjusted for (e.g. Thatcher, Walker, and Siudice, 1987).

3. Proposed Algorithm

We propose to approximate the probability in (1) by 2-dimensional mixtures of products of k univariate normal interval probabilities

$$\int_0^{\infty}\int_{-\infty}^{+\infty} \prod_{i=1}^{k} [\Phi((\sqrt{c}\lambda_i z + |q|s)/(1-c\lambda_i^2)^{1/2}) - \Phi((\sqrt{c}\lambda_i z - |q|s)/(1-c\lambda_i^2)^{1/2})]\, d\Phi(z)\, d\Gamma(s) \quad (2)$$

where $c = \pm 1$, Φ is the standard normal distribution, Γ is the distribution of s/σ, and $\lambda_1, \cdots, \lambda_k$ are constants that depend on $\mathbf{X}$. If again m-point quadrature is employed, then (2) requires $2km^2$ evaluations of Φ which is much less than the $2m^{k+1}$ evaluations required for iterated multivariate integration. Note also that $2km^2$ grows *linearly* (as opposed to *exponentially* for $2m^{k+1}$) in model size k. Thus, using this approximation, simultaneous confidence intervals in GLM can be given in an interactive environment.

If the correlation matrix $\mathbf{R}$ of $\hat{\underline{\beta}}$ satisfies

$$\mathbf{R} = \{\rho_{ij}\} = \begin{pmatrix} (1-c\lambda_1^2) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (1-c\lambda_k^2) \end{pmatrix} + c \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} (\lambda_1 \cdots \lambda_k) \quad (3)$$

with $c = +1$ (called structure-L, see Tong, 1979), then

$$\begin{pmatrix} (\hat{\beta}_1-\beta_1)/\sqrt{v_{11}} \\ \cdot \\ \cdot \\ \cdot \\ (\hat{\beta}_k-\beta_k)/\sqrt{v_{kk}} \end{pmatrix}/\sigma \equiv \begin{pmatrix} (1-\lambda_1^2)^{1/2}\, Z_1 \\ \cdot \\ \cdot \\ \cdot \\ (1-\lambda_k^2)^{1/2}\, Z_k \end{pmatrix} + \begin{pmatrix} \lambda_1 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_k \end{pmatrix} Z_0 \qquad (4)$$

where $Z_1, \cdots, Z_k, Z_0$ are *iid* N(0,1) random variables and $\equiv$ denotes equality in distribution, and it is well known (e.g. Gupta 1963) that the probability in (1) can be written as (2). Nelson (1982) shows that when (3) holds with $c = -1$, (2) is still a valid expressions for the probability in (1) provided one defines

$$\Phi(a+ib) = (2\pi)^{-1} \int_{-\infty}^{a} e^{-(u-ib)^2/2} du. \qquad (5)$$

Note that the real part of (5) is an even function of b while the imaginary part of (5) is an odd function of b, so the inner integral of the imaginary part of the integrand in (2) is zero.

While textbook ANOVA examples and response surface examples tend to have highly patterned design matrix **X**, leading to correlations **R** satisfying (3), the same cannot be said about real-life experiments: the data may be observational; the experimenter may not follow a textbook design; there may be covariates, or missing values. Our proposal is as follows. Given **R**, find the $\mathbf{R}_1$ satisfying (3) "closest" to **R**. Then solve (1) with $\mathbf{R}_1$ in place of **R**, i.e., evaluate (2) using the $\lambda_1, \cdots, \lambda_k$ of $\mathbf{R}_1$ to obtain a critical value $|q_1|$ as an approximation to $|q|$.

4. Utilizing Factor Analysis Algorithms

The implementation of our proposal depends crucially on the recognition that the problem of finding the $\mathbf{R}_+$ with $c = +1$ "closest" to **R** is the *Factor Analysis* problem of computing the "population" correlation matrix $\mathbf{R}_+$ of a 1-factor model (3) that best fit the "sample" correlation matrix **R**. (In Factor Analysis, $\lambda = (\lambda_1, \cdots, \lambda_k)'$ is referred to as the factor pattern.) Different measures of closeness correspond to different methods of Factor Analysis. For example, the Iterated Principal Factor method finds the $\mathbf{R}_+$ that minimizes $\|\mathbf{R} - \mathbf{R}_+\|$ where $\|\cdot\|$ is the Euclidean norm defined by $\|\mathbf{A}\| = (\sum_i \sum_j |a_{ij}|^2)^{1/2}$. The Maximum Likelihood method finds the $\mathbf{R}_+$ that minimizes $\text{trace}(\mathbf{R}_+^{-1}\mathbf{R}) - \log(|\mathbf{R}_+^{-1}\mathbf{R}|)$. The Generalized Least Square method finds the $\mathbf{R}_+$ that minimizes $\text{trace}((\mathbf{R}\mathbf{R}_+^{-1} - \mathbf{I})^2)$. So even though the matrix **R** is deterministic in our setting, we can use existing Factor Analysis algorithms to compute $\mathbf{R}_+$.

Given **R**, it is possible that a $\mathbf{R}_-$ with $c = -1$ can come closer to **R** than any $\mathbf{R}_+$ with $c = +1$. Structure (3) with $c = -1$ has no meaning in the usual Factor Analysis setting. However, it is well known that **R** has structure (3) with $c = 1$ if and only if $\mathbf{R}^{-1}$ has structure (3) with $c = +1$. (see Graybill, 1983, p.189) Therefore, one can find the $\mathbf{R}_-$ closest to $\mathbf{R}^{-1}$, calculate $\mathbf{R}_- = \mathbf{R}_-^{-1}$ which satisfies (3) with $c = 1$, and let $\mathbf{R}_1$ be either $\mathbf{R}_+$ or $\mathbf{R}_-$ whichever comes closer to **R**.

When **R** satisfies (3), usually the Factor Analysis algorithms will recover the correct λ_i's so the critical value given by our algorithm is theoretically exact (i.e., $|q_1| \equiv |q|$). It is also true that $|q_1| \equiv |q|$ for *any* **R** with $k \le 3$ because every **R** with $k \le 3$ satisfies (3). The case $k = 2$ is trivial. For $k \le 3$, the group of sign changes on λ_i when multiplied by $c = \pm 1$ generate all possible sign patterns of ρ_{ij}, $i \ne j$. Thus, by taking logarithms of $|\rho_{ij}|$, the $|\lambda_i|$'s can be solved as three unknowns in three linear equations, and then the proper signs can be attached. As **R** departs from (3), because of the continuous nature of our strategy, graceful degradation of the approximation $|q_1|$ can be expected.

Given a data set, the correlation matrix **R** can be obtained by applying any suitable software package to the applicable model. Then λ can be computed from commonly accessible Factor Analysis algorithms. To solve for $|q_1|$, a *q1* subroutine has been written which, given λ and degrees of freedom ν, integrates the outside integral by 24-point Gauss-Legendre quadrature, the inside integral by 24-point Gauss-Hermite quadrature, and solves for $|q_1|$ by the modified secant method. The entire process has been automated in the S® environment for the regression setting. Accessing the **R** in the structure returned by the *regress* function, a *factor* function has been written which returns λ by calling subroutine FACTR of IMSL. Accessing $\hat{\beta}$, λ, MSE, and ν returned by *regress* and *factor*, a *glmci* function has been written which calls *q1* and returns a structure defining the simultaneous confidence intervals for β. A graphics function *ci* has been written to plot these confidence intervals.

Examples of how well the proposed algorithm performs in a variety of settings are provided below.

5. Regression

Orthogonal designs clearly satisfy (3) with $\lambda_i \equiv 0$ and can be thought of as 0-factor designs in our setting. However, if observations from more than one design point are missing, or if an orthogonal design is augmented by more than one additional design point, then generally (3) is not satisfied exactly. But the approximation $|q_1|$ is often real close to $|q|$, as the following example with observational data demonstrates.

5.1 Motor vehicle death example

Data on page 191 of Draper and Smith (1981) gives, for the 49 contiguous states, number of motor vehicle deaths (Y) in 1964, number of drivers $\times 10^{-4}$ in 1964 (X_1), number of persons per square mile (X_2) in 1963, rural road mileage $\times 10^{-3}$ (X_3) in 1963, and normal January maximum temperature (X_4). The linear model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$ produces an R^2 of 0.9654 with the following correlation matrix for $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4$

$$\begin{pmatrix} 1 & - & - & - \\ 0.5638005 & 1 & - & - \\ -0.6676538 & 0.6067692 & 1 & - \\ -0.2990695 & 0.2081862 & 0.2096254 & 1 \end{pmatrix}$$

Using an Iterated Principal Factor algorithm, λ is found to be $(-0.8161890, 0.7125773, 0.8217566, 0.3053787)'$, leaving a residual correlation matrix of

$$\begin{pmatrix} 0 & - & - & - \\ 0.01779729 & 0 & - & - \\ 0.003054976 & 0.02120399 & 0 & - \\ -0.04982275 & -0.00941977 & -0.04132158 & 0 \end{pmatrix}$$

Based on this 1-factor approximation, critical values $|q_1|$ for various α are then computed. To check the accuracy of these approximate critical values, 40000 pairs of $\max_{i=1,\cdots,k} |\hat{\beta}_i - \beta_i| / s\sqrt{v_{ii}}$ were generated based on $\mathbf{R}$ and $\mathbf{R_1}$ in such a way that a control-variate variance reduction technique could be applied to reduce the standard deviation of the estimate of true α roughly by a factor of 3. We found

Nominal α	$\|q_1\|$	Unb. Est. of True α	95% CI of True α
0.10	2.229697	0.10 + 0.00006	(0.0993, 0.1009)
0.05	2.536990	0.05 + 0.00000	(0.0494, 0.0506)
0.01	3.168086	0.01 + 0.00002	(0.0097,0.0103)

It would seem that the 1-factor approximation is adequate for practical purposes. The following table compares the critical values associated with the various methods.

	Scheffé	Bonferroni	Sidak	Factor Analysis
$\alpha = 0.10$	2.883	2.321	2.305	2.230
$\alpha = 0.05$	3.215	2.605	2.597	2.537
$\alpha = 0.01$	3.888	3.207	3.206	3.168

6. One-way Designs with A Covariate

Assume the model

$$Y_{ia} = \mu_i + \beta(X_{ia} - \bar{X}_{..}) + e_{ia}, \quad i = 1,\cdots,k, \; a = 1,\cdots,n_i,$$

where $\mu_1, \cdots, \mu_k$ are the treatment effects, β is the common slope for the covariate X, $\bar{X}_{..} = \Sigma\Sigma X_{ia}/\Sigma n_{ia}$, and e_{ia} are *iid* $N(0,\sigma^2)$ with σ^2 unknown.

It can be verified that if the treatment effects $\mu_1, \cdots, \mu_k$ themselves are of interest, then (3) holds so exact simultaneous confidence intervals can be computed by our method. If, on the other hand, treatments versus control effects $\mu_1-\mu_k, \cdots, \mu_{k-1}-\mu_k$ are of interest, then the critical value $|q|$ has to be approximated. The following example demonstrates both possibilities.

6.1 Starch example

Scheffé (1959, p.216) gives breaking strength (y) in grams and thickness (x) in 10^{-4} inch from tests on 7 types of starch film. (Starch 1 = Canna, 2 = Sweet Potato, 3 = Corn, 4 = Rice, 5 = Dasheen, 6 = Wheat, 7 = Potato) It is assumed that the regression coefficient of y on x is the same for all starches.

First suppose the μ_i's themselves are of interest. Using Factor Analysis algorithms, the λ_i's in (3) for the correlation matrix $\mathbf{R}$ of $(\hat{\mu}_1, \cdots, \hat{\mu}_7)'$ are found to be

$$\begin{pmatrix} \lambda_1 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_7 \end{pmatrix} = \begin{pmatrix} 0.5782180 \\ 0.2092954 \\ -0.4571517 \\ -0.0483264 \\ -0.3582556 \\ -0.7636223 \\ 0.8210369 \end{pmatrix}$$

Exact critical values based on these λ_i's and 86 d.f. for MSE for this data are then computed by our algorithm, which compare with critical values from other methods as follows.

	Scheffé	Bonferroni	Sidak	Exact
$\alpha = 0.10$	3.538	2.501	2.484	2.449
$\alpha = 0.05$	3.851	2.756	2.749	2.721
$\alpha = 0.01$	4.470	3.296	3.294	3.283

Next suppose treatments versus control are of interest, with Potato as the control. The BLUE for the differences of breaking strength are as follows. (For later reference, the estimates employed by the MEANS option of PROC GLM in SAS, which do not take the covariate into account, are also displayed.)

Breaking Strength	With Covariate	Without Covariate
Starch 7 – Starch 1	58.94	181.14
Starch 7 – Starch 2	71.36	265.43
Starch 7 – Starch 3	119.01	493.60
Starch 7 – Starch 4	146.08	437.25
Starch 7 – Starch 5	174.90	563.74
Starch 7 – Starch 6	180.59	667.68

The correlation matrix $\mathbf{R}$ of the BLUE (with covariate) for the differences of breaking strength is

$$\begin{pmatrix} 1 & - & - & - & - & - \\ 0.39585 & 1 & - & - & - & - \\ 0.56777 & 0.49365 & 1 & - & - & - \\ 0.54682 & 0.46214 & 0.75986 & 1 & - & - \\ 0.51409 & 0.44881 & 0.76754 & 0.69301 & 1 & - \\ 0.55055 & 0.49229 & 0.86517 & 0.77387 & 0.79156 & 1 \end{pmatrix}$$

Using a maximum likelihood factor analysis algorithm, the closest 1-factor model is found to be

$$\begin{pmatrix} 0.79025\, Z_1 \\ 0.84280\, Z_2 \\ 0.38954\, Z_3 \\ 0.55876\, Z_4 \\ 0.54342\, Z_5 \\ 0.35255\, Z_6 \end{pmatrix} + \begin{pmatrix} 0.61778 \\ 0.53822 \\ 0.92101 \\ 0.82933 \\ 0.83946 \\ 0.93579 \end{pmatrix} Z_0$$

which leaves a residual correlation matrix of

$$\begin{pmatrix} 0 & - & - & - & - & - \\ 0.06604 & 0 & - & - & - & - \\ 0.00339 & -0.00206 & 0 & - & - & - \\ 0.03862 & 0.01577 & -0.00397 & 0 & - & - \\ -0.00032 & -0.00301 & -0.00562 & -0.00319 & 0 & - \\ -0.02289 & -0.01138 & 0.00329 & -0.00222 & 0.00600 & 0 \end{pmatrix}$$

with an average root mean square off-diagonal residual of 0.02143014.

Based on this 1-factor approximation, critical values $|q_1|$ for various α are then computed. To check the accuracy of these approximate critical values, again 40000 pairs of $\max_{i=1,\cdots,k} |\hat{\beta}_i - \beta_i| / s\sqrt{v_{ii}}$ were generated based on $\mathbf{R}$ and $\mathbf{R_1}$ and a control-variate technique was applied to reduce the variance of the estimate of true α. We found

Nominal α	$\|q_1\|$	Unb. Est. of True α	95% CI of True α
0.10	2.264235	0.10 – 0.000025	(0.0991, 0.1008)
0.05	2.560634	0.05 + 0.000175	(0.0496, 0.0508)
0.01	3.154060	0.01 – 0.000125	(0.0096, 0.0102)

It would seem that the 1-factor approximation is adequate for practical purposes.

The following table compares the confidence intervals obtained by various methods.

Breaking Strength	Factor Analysis CI	Sidak's CI	Scheffé's CI	SAS (Tukey) CI
Starch 7 – Starch 1	58.94 ± 127.60	58.94 ± 134.22	58.94 ± 181.29	181.14 ± 139.05
Starch 7 – Starch 2	71.36 ± 193.98	71.36 ± 204.04	71.36 ± 275.61	265.43 ± 209.73
Starch 7 – Starch 3	119.01 ± 183.67	119.01 ± 193.20	119.01 ± 260.96	493.60 ± 126.00
Starch 7 – Starch 4	146.08 ± 167.47	146.08 ± 176.16	146.08 ± 237.94	437.25 ± 142.29
Starch 7 – Starch 5	174.90 ± 207.67	174.90 ± 217.81	174.90 ± 294.21	563.74 ± 161.81
Starch 7 – Starch 6	180.59 ± 220.54	180.59 ± 231.98	180.59 ± 313.35	667.68 ± 123.13

Note that whereas all the confidence intervals based on BLUEs cover 0, none of the SAS confidence intervals covers 0. SAS computes confidence intervals with the covariate "thickness" ignored. As Figure 1 shows, the significant differences in "starch" detected by SAS are due mainly to the differences in "thickness" associated with "starch."

6.2 A simulation study

A small simulation was performed to see if the close approximation to **R** by $\mathbf{R}_1$ above for the real data example was a *fluke*. Taking $k = 10$, $n_1 = \cdots = n_{10} = 10$, and $X_1, \cdots, X_{10}$ to be *iid* standard normal, 100 correlation matrices **R** of $(\hat{\mu}_1-\hat{\mu}_k, \cdots, \hat{\mu}_{k-1}-\hat{\mu}_k)'$ were generated using PROC MATRIX in SAS. First, each **R** was checked to see if it satisfied (3). None did. Then for each **R**, the Maximum Likelihood method in PROC FACTOR of SAS was used to find the closest $\mathbf{R}_1$, and the root-mean-square of the off-diagonal elements of $\mathbf{R} - \mathbf{R}_1$ was recorded. The mean of the 100 root-mean-squares was 0.004359595, and the standard deviation was 0.002385. Their stem-and-leaf plot is given below.

N = 100 Median = 0.00393708
Quartiles = 0.002674504, 0.00537431

Decimal point is 3 places to the left of the colon

```
1 : 124456679
2 : 00012333334446667779
3 : 1122234444566777788999
4 : 000223445566677899
5 : 000113556678899
6 : 014789
7 : 26
8 : 0001
9 : 58

High: 0.01429412 0.01523449
```

7. Two-way Design with Missing Observations

Consider the two-way no-interaction model

$$Y_{ihr} = \mu + \tau_i + \gamma_h + e_{ihr},$$

$$i = 1,\cdots,a,\ h = 1,\cdots,b,\ r = 1,\cdots,n_{ih},$$

where Y_{ihr} are the observations, $\tau_1, \cdots, \tau_a$ are the treatment effects, $\gamma_1, \cdots, \gamma_b$ are the block effects, e_{ihr} are *iid* Normal$(0,\sigma^2)$, and $\tau_1-\tau_a, \cdots, \tau_{a-1}-\tau_a$ are of interest. If the cell sizes n_{ih} are proportionate, that is, $n_{ih} = w_i m_h$ for all i and h, then (3) is satisfied, but not generally otherwise. However, recall (3) is always satisfied when $k \le 3$, and the following example illustrates this.

7.1 Blood example

We use a data set popular for illustrating two-way unbalanced ANOVA (Fleiss, 1986, p.166; *SAS User's Guide: Statistics*, 1985, p.492). Of interest was the increase in systolic blood pressure in dogs after *treatment*, with *disease* as a blocking factor. The sample means and sample sizes (in parentheses) are given in the following table.

	Treatment			
Disease	1	2	3	4
1	29.333 (6)	28.000 (8)	16.333 (3)	13.600 (5)
2	28.250 (4)	33.500 (4)	4.400 (5)	12.833 (6)
3	20.400 (5)	18.167 (6)	8.500 (4)	14.200 (5)

Using a Factor Analysis algorithm, λ is found to be (0.69567,0.69901,0.64584)', from which exact critical values were computed. The following table compares the critical values given by the various methods.

	Scheffé	Bonferroni	Sidak	Exact
$\alpha = 0.10$	2.565	2.187	2.172	2.119
$\alpha = 0.05$	2.889	2.474	2.468	2.427
$\alpha = 0.01$	3.542	3.077	3.076	3.052

7.2 A simulation study

A small simulation was performed to see how well can $\mathbf{R}_1$ approximate **R** for larger k. In particular, we took a = b = 10, $n_{11} = \cdots = n_{10,10} = 10$, and consider the *missing completely at random* case (c.f. Little and Rubin, 1987, p.14), where independently each observation has a 0.5 probability of missing. Using PROC MATRIX in SAS, 100 correlation matrices **R** of $(\hat{\tau}_1-\tau_a, \cdots, \tau_{a-1}-\tau_a)'$ were generated. First, each **R** was checked to see if it satisfied (3). None did. Then for each **R**, the Maximum Likelihood method in PROC FACTOR of SAS was used to find the closest $\mathbf{R}_1$, and the root-mean-square of the off-diagonal elements of $\mathbf{R} - \mathbf{R}_1$ was recorded. The mean of the 100 root-mean-squares was 0.001343971, and the standard deviation was 0.00030142. Their stem-and-leaf plot is given below.

N = 100 Median = 0.00134159
Quartiles = 0.00112995, 0.001558005

Decimal point is 4 places to the left of the colon

```
6 : 25
7 : 9
8 : 05
9 : 13889
10 : 0123445678
11 : 01133334667899
12 : 0012333457889
13 : 12355567789
14 : 11122356789999
15 : 3357889
16 : 0012366
17 : 05567
18 : 223577
19 : 1
20 : 4
21 : 0
```

8. Vector Processing

It was relatively straight forward to code the proposed algorithm so that it will take advantage of vector processing capability when available. On a Cray X-MP with the *cft77* compiler for example, execution is at least five times faster with the vector processing capability turned on as compared to optionally using the scalar processing capability only.

Acknowledgement

Research was supported in part by Grant No. 5 R01 CA41168, awarded by the National Cancer Institute. The author thanks Ms. Bekka Denning for technical advice. Prof. Angela Dean first expressed doubt concerning the MEANS option of PROC GLM to the author. Prof. Sue Leurgans pointed out the references [3] and [11]. Computing resources were provided by the Mathematical Sciences Computing Laboratory of the Ohio State University, and the Ohio Supercomputing Center.

References

[1] Draper, N. R. and Smith, H. (1981). *Applies Regression Analysis, 2nd Edition*, Wiley, New York.

[2] Fleiss, J. (1986). *The Design and Analysis of Clinical Experiments*, Wiley, New York.

[3] Fuchs, C. and Sampson, A. (1987). Simultaneous confidence intervals for the General Linear Model. *Biometrics* 43, 457-469.

[4] Graybill, F. A. (1983). *Matrices with Applications in Statistics, 2nd Edition*. Wadsworth, CA.

[5] Gupta, S. S. (1963). Probability integrals of multivariate normal and multivariate *t*. *Annals of Mathematical Statistics* 34, 792-828.

[6] Little, R. J. A. and Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. Wiley, New York.

[7] Nelson, P. R. (1982). The multivariate normal distribution with multiplicative correlation structure. Technical report, Department of Statistics, The Ohio State University.

[8] *SAS User's Guide: Statistics, Version 5 Edition* (1985). SAS Institute, Inc., Cary, NC.

[9] Schervish, M. J. (1984). Multivariate normal probability with error bound. Algorithm AS 195, *Applied Statistics* 33, 81-94.

[10] Tong, Y. L. (1980). *Probability Inequalities in Multivariate Distributions*. Academic Press, New York.

[11] Thatcher, R.W., Walker, R.A., Siudice, S. (1987). Human cerebral hemispheres develop at different rates and ages. *Science* 236, 1110-1113.

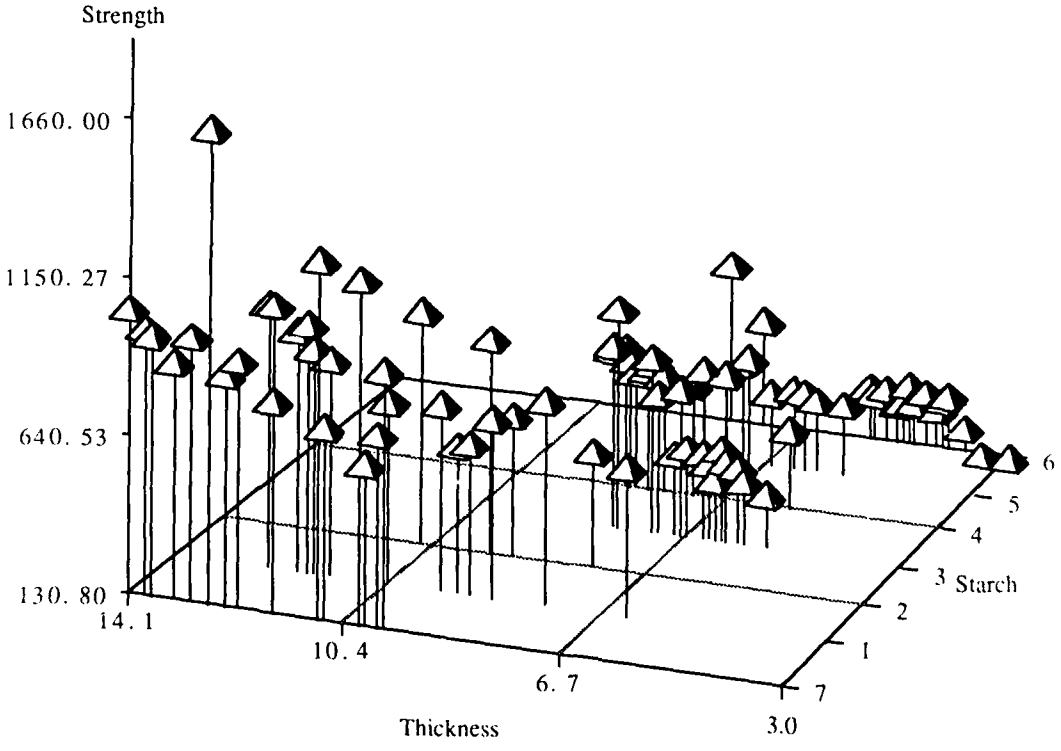


Figure 1

ASSESSMENT OF PREDICTION PROCEDURES IN MULTIPLE REGRESSION ANALYSIS

Victor Kipnis, University of Southern California

1. Introduction. As opposed to the traditional inference based on a priori specified model, the main feature of the modern regression analysis is model building with regard to some specified regression goals. This process usually involves some scrutinizing of both the available data and a set of potential equations before settling on the final version of the model. Such examples of this strategy as examination of residuals, checking standard assumptions of homoscedasticity and of absence of cerial correlation, analysis of outliers, regression diagnostics, sorting out, or transforming data, choosing form of the equation, selecting explanatory variables, etc., constitute what is usually called exploratory, or else, data-analytical approach to modelling. Exploratory methods are often used iteratively and in alternation with fitting of tentative models and, thus, make contemporary regression analysis a complex, multistep, iterative process. In practice, when all this activity is carried out using and reusing the same data, conventional means of inference both at the intermediate steps and for the finally chosen model could be misleading, sometimes badly so (e.g. Freedman, 1983; Lovell, 1983; Miller, 1984; Pinsker et al., 1985, 1987).

This paper concentrates on the evaluation of exploratory procedures for prediction, which is one of the major goals in applied regression analysis. In this case model building is usually reduced to selection of the 'most efficient' predictor among the class of potential regression equations, that is, one that provides the minimum mean squared error of prediction (MSEP). There is a good deal of literature on different selection procedures with regard to their computational (logical) scheme (e.g. see a review in Hocking, 1976), but much less has been done concerning analysis of statistical properties of thereby selected predictors. As was repeatedly pointed out (e.g. Berk, 1978; Hjorth, 1982; Miller, 1984) the theory behind the conventional MSEP estimators is not valid when predictor selection and estimation are from the same data. The very selection process introduces considerable distortion into the distribution of these estimators and, in particular, leads to their substantial bias when the selection effect is not allowed for. The present paper attacks this problem along the same lines as in (Kipnis, Pinsker, 1983; Pinsker et al., 1985; Kipnis, 1987).

We bring in the 'procedural approach' and suggest that assessment of the efficiency of any predictor should rest on the assessment of the procedure by which this predictor has been chosen, rather than the evaluation of any particular prediction equation. As exact distributional results are virtually impossible to obtain, even for relatively simple selection procedures, it is suggested to estimate procedural performance in a simulation study by generating bootstrap pseudosamples and applying to them the same regression procedure that was used for the original data.

For illustrative purposes, prediction procedures based on subset selection methods are considered. It should be emphasized, though, that the general concepts of the present study apply to any other data-analytical procedures for model building.

2. Problem Formulation. Procedural Approach. Consider the linear regression model

$$Y = Y^0 + \epsilon = X\beta + \epsilon, \qquad (1)$$

where $Y = (Y^1, \ldots, Y^n)'$ is a n-vector of observations on the response variable, $X = [X_1, \ldots, X_k]$ is $n \times k$ full rank matrix of observations on each of the k predictor variables, $\beta = (\beta^1, \ldots, \beta^k)'$ is a k-vector of unknown coefficients, $\epsilon = (\epsilon^1, \ldots, \epsilon^n)'$ is an n-vector of unobservable disturbances, $\epsilon \sim N(0, \sigma^2 I_n)$. The X's are assumed fixed.

The given $n \geq k$ observations on the response and the predictor variables constitute, we suppose, a construction set $V = (X_V, Y_V)$. Let $W = (X_W, Y_W)$ be a new set of n_W observations based on the same model (1):

$$Y_W = Y_W^0 + \epsilon_W = X_W\beta + \epsilon_W,$$

where ϵ_W is independent from ϵ_V. We will call W the 'target' set. Given the construction set V and the matrix X_W, the regression goal is to predict vector Y_W with some predictor $\hat{Y}_W$ based on a p-subset, $p \leq k$, of the predictor variables. We assume that this predictor is selected among the class of potential predictors (say, all possible subsets) by applying to the data V some prediction procedure g.

Without providing any strict formalization, by 'procedure' we will understand a mapping from the input data to the output. In the present case the input consists of the construction set V, matrix X_W, the class of potential predictors, and, perhaps, some criteria for choosing a particular subset and estimating the corresponding parameters. The output includes the chosen subset of predictor variables, estimates of its parameters, and vector $\hat{Y}_W$. Roughly speaking, all operations performed on the construction set V to get the output, the entire exploratory process, is covered by the use of the term procedure.

It should be stressed that procedure g is conceived of as a separate whole, a distinct statistical entity, in spite of the fact that various nonstatistical considerations (experience, professional intuition, simplicity of computation, etc.) may and in fact do influence its choice. The relationship between procedure for model building and built up model reminds one between estimator and estimate. Just like estimate is simply a number, a selected model is characterized by a realized vector of estimated parameters. Its justification should rest on an assessment of the procedure which, by analogy with estimator, is the 'recipe' or 'selection strategy', or 'algorithm' by which the data are transformed into an actual model. As a result of such conceptualization, procedure becomes an object of statistical study, allows statistically valid evaluation and comparison of different procedures, and, thus, belongs to the sphere of formal inference.

Turning back to subset selection, a typical prediction procedure selects a p-subset $X_p = [X_{i1}, \ldots, X_{ip}]$ and yields the n_W-vector $\hat{Y}_W = X_W\hat{\beta}$ based on the OLS fitting of the selected subset. If the full set of regressors is not selected, some of the components of the realized vector $\hat{\beta}$ are zero.

It is important to emphasize that the chosen predictor $\hat{Y}_W$ does not pretend to represent the 'real' model by including all the significant variables and excluding all the nonsignificant ones (with $\beta_j = 0$). Moreover, procedure g selects a random subset of predictor variables which may vary for different realizations of V. Because of that

fact, evaluation of a predictor $\hat{Y}_W = g(V; X_W)$ should be based on the assessment of the selection procedure g rather than any particular model (selected subset) chosen for the given realization of the construction set.

Consider prediction error

$$e_W = \hat{Y}_W - Y_W = g(V; X_W) - Y_W \tag{2}$$

Since X_V and X_W are assumed fixed and known, below we will use notations $\hat{Y}_W = g_W(Y_V)$. Let us split e_W into a nonrandom and a random parts:

$$e_W = e^0_W + \delta e_W,$$

where

$$e^0_W = \hat{Y}^0_W - Y^0_W = g_W(Y^0_V) - Y^0_W \tag{3}$$

is an error of predicting the nonrandom part Y^0_W of Y_W by applying procedure g to the nonrandom part $V^0 = (X_V, Y^0_V)$ of the construction set V. From (2)-(3) it follows

$$\delta e_W = e_W - e^0_W = \delta g_W - \epsilon_W \tag{4}$$

where $\delta g_W = g_W(Y_V) - g_W(Y^0_V)$. For the quadratic loss function

$$L_W(g; V) = \frac{1}{n_W} e'_W e_W \tag{5}$$

two distinct risk functions are important measures of efficiency of the procedure g. The first one is the conditional MSEP for a fixed vector Y_V (or the fixed construction set V, since X_V is fixed anyway).

$$R(g; Y_V) = MSEP(g_W(Y_V) \mid Y_V) = E[L_W(g; V) \mid V]$$

It follows from (2)-(5) that

$$R(g; Y_V) = \sigma^2 + \frac{1}{n_W}(g_W(Y_V) - Y^0_W)'(g_W(Y_V) - Y^0_W) \tag{6}$$

Note, that $R(g; Y_V)$ is a random variable together with Y_V and could be considered as measuring both the predictive ability of a selected model and the efficiency of the selection procedure g for a given construction set V.

To be able to analyze existing procedures and to invent new ones, one needs to know their statistical properties for different realizations of V. One such characteristic is the unconditional MSEP, i.e., the average risk over all possible construction sets:

$$\begin{aligned} R(g) &= E_{Y_V}[R(g; Y_V)] \\ &= \sigma^2 + \frac{1}{n_W} E e'_W E e_W + \frac{1}{n_W} tr(VAR[\delta g_W]), \end{aligned} \tag{7}$$

where $VAR[\delta g_W] = E[(\delta g_W - E\delta g_W)(\delta g_W - E\delta g_W)']$ is the variance - covariance matrix of δg_W. As opposed to the conditional risk (6), the unconditional risk (7) measures the average efficiency of the selection procedure g, not of a selected model, which depends on the realization of V.

There is no full agreement in the literature on whether conditional or unconditional MSEP is more appropriate for prediction assessment. The traditional statistics have been originally proposed as estimators of the unconditional MSEP (e.g., Hocking, 1976), but they seem now to be looked upon as estimators of the conditional risk (7) (see, Hjorth, 1982; Efron, 1983; Picard and Cook, 1984). Below we will consider statistical properties of different estimators with regard to both measures (6) and (7).

The most obvious estimator of MSEP is the apparent losses or autolosses

$$\begin{aligned} AL(g; V) &= \frac{1}{n}(\hat{Y}_V - Y_V)'(\hat{Y}_V - Y_V) \\ &= \frac{1}{n}(g_V(Y_V) - Y_V)'(g_V(Y_V) - Y_V) \end{aligned} \tag{8}$$

that measure goodness-of-fit of the procedure g on the construction set V. For most procedures AL tends to underestimate MSEP, because the same data have been used for both construction and evaluation. This is a familiar fact that could be easily demonstrated when the procedure g consists of OLS fitting of an a priori specified subset X_{Vp}, and $X_W = X_V$, i.e., when we predict new observations $Y_W = X_V\beta + \epsilon_W$ for the same set X_V of explanatory variables. For future references we will denote this procedure by g_p. We have

$$g_p(Y_V) = P_p Y_V, \tag{9}$$

where $P_p = X_{Vp}(X'_{Vp}X_{Vp})^{-1}X'_{Vp}$ is the projection matrix onto the linear space spanned by the column-vectors of the matrix X_{Vp}. In this case $Y^0_W = X_V\beta = Y^0_V$, $e^0_W =$

Note, that $R(g; Y_V)$ is a random variable together with Y_V and could be considered as measuring both the predictive ability of a selected model and the efficiency of the selection procedure g for a given construction set V.

To be able to analyze existing procedures and to invent new ones, one needs to know their statistical properties for different realizations of V. One such characteristic is the unconditional MSEP, i.e., the average risk over all possible construction sets:

$$\begin{aligned} R(g) &= E_{Y_V}[R(g; Y_V)] \\ &= \sigma^2 + \frac{1}{n_W} E e'_W E e_W + \frac{1}{n_W} tr(VAR[\delta g_W]), \end{aligned} \tag{7}$$

where $VAR[\delta g_W] = E[(\delta g_W - E\delta g_W)(\delta g_W - E\delta g_W)']$ is the variance - covariance matrix of δg_W. As opposed to the conditional risk (6), the unconditional risk (7) measures the average efficiency of the selection procedure g, not of a selected model, which depends on the realization of V.

There is no full agreement in the literature on whether conditional or unconditional MSEP is more appropriate for prediction assessment. The traditional statistics have been originally proposed as estimators of the unconditional MSEP (e.g., Hocking, 1976), but they seem now to be looked upon as estimators of the conditional risk (7) (see, Hjorth, 1982; Efron, 1983; Picard and Cook, 1984). Below we will consider statistical properties of different estimators with regard to both measures (6) and (7).

The most obvious estimator of MSEP is the apparent losses or autolosses

$$\begin{aligned} AL(g; V) &= \frac{1}{n}(\hat{Y}_V - Y_V)'(\hat{Y}_V - Y_V) \\ &= \frac{1}{n}(g_V(Y_V) - Y_V)'(g_V(Y_V) - Y_V) \end{aligned} \tag{8}$$

that measure goodness-of-fit of the procedure g on the construction set V. For most procedures AL tends to underestimate MSEP, because the same data have been used for both construction and evaluation. This is a familiar fact that could be easily demonstrated when

the procedure g consists of OLS fitting of an a priori specified subset X_{Vp}, and $X_W = X_V$, i.e., when we predict new observations $Y_W = X_V\beta + \epsilon_W$ for the same set X_V of explanatory variables. For future references we will denote this procedure by g_p. We have

$$g_p(Y_V) = P_p Y_V, \tag{9}$$

where $P_p = X_{Vp}(X'_{Vp}X_{Vp})^{-1}X'_{Vp}$ is the projection matrix onto the linear space spanned by the column-vectors of the matrix X_{Vp}. In this case $Y^0_W = X_V\beta = Y^0_V$, $e^0_W = e^0_V$, $E(g_p(Y_V)) = P_pY^0_V = g_p(Y^0_V)$, and it follows from (7)-(8) that

$$R(g_p) = \frac{1}{n}e^0_V{}'e^0_V + \frac{\sigma^2}{n}(n+p) \tag{10}$$

and

$$AR(g_p) = E\big[AL(g_p;V)\big] = \frac{1}{n}e^0_V{}'e^0_V + \frac{\sigma^2}{n}(n-p), \tag{11}$$

so that AL underestimates MSEP with the negative bias $-2p\sigma^2/n$.

Such 'adjusted' forms of AR as J_p (Rothman, 1968), C_p (Mallows, 1973), AIC (Akaike, 1973), and other conventional estimators have been suggested to allow for this bias or 'overoptimism' of the self-evaluating autoloss function. These statistics are asymptotically unbiased under certain conditions, the major assumption being that g is based on fitting a subset that has been chosen independently from the construction data V. Thus, conventional estimators only partly adjust AL with regard to self-evaluation, while the real subset selection has never been allowed for. As a result, these estimators still carry some overoptimism and, in their turn, need adjustment. To be able to get more adequate estimators of the prediction risk, one has to study the distribution of $\hat{Y}_W$ under that very procedure g which has yielded this predictor.

3. Pseudosample Method. The idea is to analyze statistical properties of predictor $\hat{Y}_W$ by applying procedure g to data generated by a known random mechanism. The main requirement is that this mechanism, or as we will call it, pseudomodel, should simulate the unknown model (1). In other words, it should generate pseudosamples that are 'close' to the real ones with regard to their statistical structure.

Consider the maximum likelihood estimator of the model (1)

$$\tilde{Y} = X\tilde{\beta} + \tilde{\epsilon}, \qquad \tilde{\epsilon} \sim N(0, \tilde{\sigma}^2 I_n), \tag{12}$$

where $\tilde{\beta} = (X'_VX_V)^{-1}X'_VY_V$ and $\tilde{\sigma}^2 = Y'_V(I_n - P_k)Y_V/(n-k)$ are the OLS estimates of the parameters based on the set of all the explanatory variables. The decision to use the 'ful' estimated model (12) and not, say, the subset selected by the procedure g is made because our goal here is simulation and not prediction. By using unbiased ML estimators of the parameters β and σ^2, we hope to get a pseudomodel that is as close to the real one as possible. Note, that this choice is not mandatory for the suggested approach, and, in principle, other pseudomodels could be used in different situations. In the present case a pseudomodel (12) is the parametric bootstrap model as described in (Efron, 1982).

Consider now a pseudoconstruction sample $\tilde{V} = (X_V, \tilde{Y}_V)$ and a pseudotarget sample $\tilde{W} = (X_W, \tilde{Y}_W)$, where $\tilde{Y}_V$ and $\tilde{Y}_W$ are, respectively, $n\times 1$ and $n_W \times 1$ random vectors independently generated from model (12). Applying the same selection procedure g that is used for the original construction set V to the pseudosample $\tilde{V}$, we get a pseudopredictor $\hat{\tilde{Y}}_W = g_{\tilde{W}}(\tilde{Y}_V)$ and a vector $\tilde{e}_W = \hat{\tilde{Y}}_W - \tilde{Y}_W$ of pseudoerrors. As pseudomodel (12) is completely known, we can, at least in principle, analyze the distribution of $\tilde{e}_W$ and use its characteristics as estimates of their counterparts for the distribution of the real vector e_W.

One possible approach to deriving pseudosample estimators is as follows. Instead of directly estimating $MSEP(g)$ with the corresponding pseudorisk, let us evaluate overoptimism of the autolosses $AL(g)$ in order to make an appropriate adjustment. At least two choices present themselves for representing average overoptimism: the difference

$$Q^A(g) = R(g) - AR(g)$$

and the ratio

$$Q^M(g) = R(g)/AR(g)$$

We will estimate $Q^A(g)$ and $Q^M(g)$ with their pseudocounterparts

$$\tilde{Q}^A(g) = \tilde{R}(g) - A\tilde{R}(g) = E_\sim\big[L_{\tilde{W}}(g,\tilde{V}) - AL(g,\tilde{V})\big]$$

and

$$\begin{aligned}\tilde{Q}^M(g) &= \tilde{R}(g)/A\tilde{R}(g)\\ &= E_\sim\big[L_{\tilde{W}}(g,\tilde{V})\big]/E_\sim\big[AL(g,\tilde{V})\big],\end{aligned}$$

where $E_\sim$ indicates expectation with respect to the random mechanism (12). In these formulas, the construction sample V is held fixed. This yields the following two pseudosample estimators of $MSEP$: the additive estimator

$$\hat{R}^A = AL(g) + \tilde{Q}^A(g)$$

and the multiplicative one

$$\hat{R}^M = AL(g)\tilde{Q}^M(g)$$

The reason behind these estimators is to get more pivotal statistics as compared to the direct estimator $\tilde{R}(g)$.

4. Linear Procedures. If procedure g is simple enough, pseudosample estimators could be studied analytically. Consider an important case when g is linear with respect to Y_V:

$$g_W(Y_V) = G_W Y_V,$$

where G_W is $n_W \times n$ matrix. As $Ee_W = e^0_W$ and $VAR[\delta g_W] = \sigma^2 G_W G'_W$, it follows from (7) - (8) that

$$R(g) = \sigma^2 + \frac{1}{n_W}e^0_W{}'e^0_W + \frac{\sigma^2}{n_W}tr(G_W G'_W) \tag{13}$$

$$AR(g) = \frac{1}{n}e^0_V{}'e^0_V + \frac{\sigma^2}{n}tr\big[(G_V - I_n)(G_V - I_n)'\big] \tag{14}$$

In a similar way, from model (12) we have

$$\tilde{R}(g) = \tilde{\sigma}^2 + \frac{1}{n_W}\tilde{e}_W^{0\prime}\tilde{e}_W^0 + \frac{\tilde{\sigma}^2}{n_W}tr(G_W G_W') \qquad (15)$$

$$A\tilde{R}(g) = \frac{1}{n}\tilde{e}_V^{0\prime}\tilde{e}_V^0 + \frac{\tilde{\sigma}^2}{n}tr\left[(G_V - I_n)(G_V - I_n)'\right], \quad (16)$$

where $\tilde{e}_V^0 = (G_V - I_n)X_V\tilde{\beta}$ and $\tilde{e}_W^0 = (G_W - I_n)X_V\tilde{\beta}$. Comparing expressions (13) - (14) and (15) - (16) we get

Theorem 1. If g is a linear procedure, and if $\tilde{\beta}$ and $\tilde{\sigma}^2$ are unbiased estimators of β and σ^2 respectively, $\hat{R}^A$ is an unbiased estimator of the prediction risk $MSEP$.

The multiplicative estimator $\hat{R}^M$ still remains biased, except for the important case when $e_V^0 = e_W^0 = 0$. As follows from the definition (3), e_W^0 represents 'bias' in applying procedure g to the nonrandom data V^0. We will call it o-bias. Any procedure that does not have o-bias is called o-adequate. o-adequacy means the property to build up 'true' model on the 'faultless' data. Putting $e_V^0, e_W^0, \tilde{e}_V^0, \tilde{e}_W^0$ to zero in formulas (13) - (16), we have

Theorem 2. If g is a linear o-adequate procedure, $\hat{R}^M$ is an unbiased estimator of the prediction risk $MSEP$.

Consider again an example of procedure g_p (9) when $X_W = X_V$. Here $G_W = G_V \equiv P_p$, so that it follows from (15)-(16) that

$$\hat{R}^A = \frac{1}{n}(RSS_p + 2p\tilde{\sigma}^2)$$

As $\tilde{\sigma}^2 = \hat{\sigma}^2$ is the mean residual sum of squares for the full model (1), $\hat{R}^A$ coincides with the traditional adjusted (unscaled) estimator of $MSEP$ (e.g. Hocking, 1976)

$$\hat{R}^{tr} = \frac{1}{n}(RSS_p + 2p\hat{\sigma}^2) = \frac{1}{n}\hat{\sigma}^2(C_p + n) \qquad (17)$$

If, in addition, the fitted subset represents the true model, and we use this information to estimate β in (12) by $\tilde{\beta} = (X_{Vp}'X_{Vp})^{-1}X_{Vp}'Y_V$, $\hat{R}^M$ coincides with another adjusted estimator

$$J_p = \frac{RSS_p}{n}\left(\frac{n+p}{n-p}\right) \qquad (18)$$

5. Nonlinear Procedures. Empirical Results. When procedure g involves selection it becomes nonlinear, and distributional results for pseudooveroptimism are virtually impossible to obtain analytically. In this case $\tilde{Q}^A$ and $\tilde{Q}^M$ must be evaluated by Monte-Carlo: independent pseudosamples $(\tilde{V}_1, \tilde{W}_1), \ldots, (\tilde{V}_N, \tilde{W}_N)$ are generated by the known pseudomodel (12), and for each $\tilde{V}_i$ the pseudopredictor $g_{\tilde{W}}(\tilde{Y}_{V_i})$ is calculated. Here X_V and X_W remain the same as in the observed data. By comparing $\hat{\tilde{Y}}_{W_i}$ and $\tilde{Y}_{W_i}$ a pseudoerror $\tilde{e}_{W_i}$ is calculated. This gives pseudoreplications $L_W(g, \tilde{V}_i) = \frac{1}{n_W}\tilde{e}_{W_i}'\tilde{e}_{W_i}$ and $AL(g, \tilde{V}_i) = \frac{1}{n}\tilde{e}_{V_i}'\tilde{e}_{V_i}$. Finally, we approximate $\tilde{Q}^A$ and $\tilde{Q}^M$ by the averages $\frac{1}{N}\sum_{i=1}^{N}\left[L_{\tilde{W}}(g, \tilde{V}_i) - AL(g, \tilde{V}_i)\right]$ and $\frac{1}{N}\sum_{i=1}^{N}L_{\tilde{W}}(g, \tilde{V}_i)/\frac{1}{N}\sum_{i=1}^{N}AL(g, \tilde{V}_i)$ respectively.

To illustrate the effect of subset selection on traditional estimators and to compare these estimators with the pseudosample ones, the following simulation study was conducted. In all the experiments the simulated data satisfy model (1), where X_V is orthonormal with $n = 50$ rows and $k = 15, 25, 35$ columns, $\sigma^2 = 1$, and $X_W = X_V$. As was pointed out in (Miller, 1984), the orthogonal case gives an example of intermediate corruption of traditional estimators under subset selection. The procedure g represents the method of all possible regressions (Hocking, 1976) and consists of screening all 2^k subsets and selecting the 'best' one with regard to the minimum C_p (or $\hat{R}^{tr}$) criterion. The two pseudosample estimators, $\hat{R}^A$ and $\hat{R}^M$, were compared with the two traditional estimators: $\hat{R}^{tr}$ (17) and J_p (18). Two values for the true vector β were considered $\beta_1 = (0, 0, \ldots, 0)'$, which represents the model with no significant variables, and $\beta_2 = (7.0, 5.0, 0, \ldots, 0)'$. The second model has two very significant predictor variables with the 'signal to noise' ratio $(\beta^i)^2/\sigma^2$ being 49 and 25 respectively. The estimators $\hat{R}^A$ and $\hat{R}^M$ were calculated by generating 100 pseudosamples for each simulated data set.

A summary of the results averaged over 400 simulated data sets is given in Tables 1 and 2. The columns 'CMSE' and 'UMSE' report the mean squared error of each estimator with regard to the conditional and unconditional risk, respectively. One can see from the Tables 1, 2 that both traditional estimators are considerably biased downward, especially for ratio k/n close to 1 ($k = 35$), when their bias reaches almost 50% of the actual $MSEP$ value. The pseudosample estimators considerably reduce bias. Moreover, contrary to the traditional estimators, they are somewhat 'pessimistic' by slightly overestimating the actual $MSEP$. On the other hand, pseudosample estimators have bigger variance than the traditional ones, so that their MSE only slightly, if any, better than MSE of $\hat{R}^{tr}$, especially when we consider conditional risk. $\hat{R}^A$ varies a little less than $\hat{R}^M$ and has the lowest MSE. J_p is the worst of all the considered estimators. With regard to the unconditional risk, both $\hat{R}^A$ and $\hat{R}^M$ outperform the traditional estimators having about 17-20% lesser MSE than $\hat{R}^{tr}$.

Although pseudosample estimators demonstrate some better results than the traditional ones, especially with regard to biasedness, they are not very impressive. One explanation is that the considered procedure g involves very extensive search. Thus, being very 'nonlinear', it becomes very sensitive to the choice of $\tilde{Y}^0 = X\tilde{\beta}$ for the pseudomodel (12). Although $\tilde{Y}^0$ is an unbiased estimator of Y^0, $E_\sim\|\tilde{Y}^0\|^2 = \|Y^0\|^2 + k\sigma^2$, so that pseudomodel (12) is based on a response vector with a much inflated length.

One possible way of coping with this difficulty is splitting a comprehensive multistep procedure into subprocedures (intermediate steps), estimating tentative results for each of them, and choosing the final predictor based on these estimates. The reason behind this approach is that subprocedures are less nonlinear and so may be less sensitive to the choice of pseudomodel (12).

Turning back to our example, consider subprocedures that for each $p = 0, 1, \ldots, k$, select the best subset with respect to C_p among all possible p-subsets. The original procedure g consists of consecutively applying these subprocedures for each p, estimating $MSEP$ for each selected subpredictor, and, finally, choosing the best one that corresponds to the minimum estimated

Table 1. Simulation Results from 400 Trials for $\beta_1 = (0, 0, \ldots, 0)'$

Estimators	$k = 15$ Mean	$k = 15$ CMSE	$k = 15$ UMSE	$k = 25$ Mean	$k = 25$ CMSE	$k = 25$ UMSE	$k = 35$ Mean	$k = 35$ CMSE	$k = 35$ UMSE
Actual $MSEP$	1.19	0	.016	1.31	0	.029	1.44	0	.042
J_p	.90	.140	.121	.83	.315	.272	.72	.622	.553
$\hat{R}^{tr}$	.91	.133	.114	.86	.281	.241	.79	.523	.462
$\hat{R}^{A}$	1.29	.124	.092	1.48	.264	.184	1.64	.502	.350
$\hat{R}^{M}$	1.30	.131	.098	1.50	.280	.202	1.67	.528	.382

Table 2. Simulation Results from 400 Trials for $\beta_2 = (7.0, 5.0, 0, \ldots, 0)'$

Estimators	$k = 15$ Mean	$k = 15$ CMSE	$k = 15$ UMSE	$k = 25$ Mean	$k = 25$ CMSE	$k = 25$ UMSE	$k = 35$ Mean	$k = 35$ CMSE	$k = 35$ UMSE
True $MSEP$	1.21	0	.016	1.33	0	.028	1.46	0	.041
J_p	.95	.130	.111	.86	.302	.259	.75	.609	.541
$\hat{R}^{tr}$	.96	.122	.103	.91	.261	.220	.81	.515	.435
$\hat{R}^{A}$	1.28	.120	.090	1.48	.258	.182	1.65	.496	.350
$\hat{R}^{M}$	1.30	.127	.096	1.50	.270	.200	1.68	.524	.381

$MSEP$. The results of the corresponding simulation experiments, based on the same model specification as above, are reported in (Kipnis, 1987). Some of them are reproduced in Table 3 that contains empirical average and CMSE for the less corrupted of the two traditional estimators, $\hat{R}^{tr}$, and pseudosample estimator, $\hat{R}^{A}$, for each of the selected subpredictors. One can see that $\hat{R}^{tr}$ is still considerably biased downward when p exceeds the true number of non-zero components of β, but less than the full size k. What is even worse, $\hat{R}^{tr}$ does not follow the actual $MSEP$. Thus, it has its minimum when $p = 4$ for $\beta = \beta_1$ and when $p = 6$ for $\beta = \beta_2$. On the contrary, as expected, the pseudosample estimator $\hat{R}^{A}$ behaves here much better than in the previous case. It not only considerably reduces bias, but also has much lesser MSE than $\hat{R}^{tr}$, especially when p is not too close to k. Moreover, it matches the actual $MSEP$ having smallest values when $p = 0$ for the first model and when $p = 2$ for the second one.

Table 4 reports the average $MSEP$ for the final predictor selected among all the best subpredictors according to criteria based on each of the five statistics: actual $MSEP, J_p, \hat{R}^{tr}, \hat{R}^{A}$, and $\hat{R}^{M}$. It follows that applying the suggested approach at the intermediate steps of model building leads to substantially better final predictor than those based on traditional criteria.

6. Conclusion. The procedural approach consists in evaluating a selected predictor by assessing the selection procedure which has yielded this predictor. For this purpose, it is suggested to construct a pseudomodel, to generate a necessary number of pseudosamples, and to apply to each one of them the same selection procedure that is used for the original data. The corresponding empirical distribution of pseudoerrors provides all characteristics of interest. This method appears general and

Table 3. Subprocedures: Simulation Results from 400 Trials; $u = 50$, $k = 25$

Subset Size p	$\beta_1 = (0, 0, \ldots, 0)'$ True MSEP	$\hat{R}^{tr}$ Mean	$\hat{R}^{tr}$ CMSE	$\hat{R}^{A}$ Mean	$\hat{R}^{A}$ CMSE	$\beta_2 = (7.0, 5.0, 0, 0, \ldots, 0)'$ True MSEP	$\hat{R}^{tr}$ Mean	$\hat{R}^{tr}$ CMSE	$\hat{R}^{A}$ Mean	$\hat{R}^{A}$ CMSE
0	1.00	1.01	.04	1.01	.04	2.48	2.46	.15	2.46	.15
1	1.11	.94	.06	1.11	.05	1.55	1.49	.08	1.55	.09
2	1.18	.91	.11	1.18	.07	1.05	1.04	.05	1.10	.06
3	1.24	.89	.16	1.23	.08	1.15	.97	.07	1.16	.07
4	1.29	.881	.20	1.27	.09	1.22	.95	.12	1.22	.08
5	1.32	.882	.24	1.30	.11	1.27	.933	.16	1.26	.10
6	1.36	.89	.27	1.33	.12	1.32	.929	.20	1.30	.11
7	1.38	.90	.29	1.36	.13	1.35	.932	.23	1.33	.12
8	1.41	.92	.30	1.38	.14	1.38	.94	.26	1.36	.13
9	1.43	.94	.31	1.39	.15	1.41	.96	.27	1.38	.14
10	1.44	.96	.31	1.41	.16	1.43	.98	.28	1.40	.15
15	1.50	1.10	.27	1.45	.19	1.49	1.11	.26	1.45	.19
20	1.51	1.28	.21	1.47	.20	1.51	1.28	.21	1.47	.20
25	1.52	1.47	.20	1.47	.20	1.52	1.47	.20	1.47	.20

Table 4. Average $MSEP$ for Predictor Selected by Different Criteria

Criterion		True MSEP	J_p	$\hat{R}^{tr}$	$\hat{R}^A$	$\hat{R}^M$
Ave $MSEP$ for	$\beta = \beta_1$	1.00	1.39	1.31	1.07	1.11
Selected predictor	$\beta = \beta_2$	1.04	1.40	1.33	1.14	1.17

flexible enough and, in principle, could be used with any selection procedure. When procedure is rather complex (e.g. includes extensive search), it becomes sensitive to the choice of an appropriate pseudomodel which should be 'close' enough to the original one. One way of coping with this problem is not to delay the assessment of the procedure until the end of the selection process, but to split the procedure into some simpler subprocedures to be able to evaluate the intermediate steps. As a result, the suggested approach can be used not only for assessing the efficiency of existing procedures, but for constructing new, more efficient procedures, as was demonstrated by the last example.

References

Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle, *2nd Intern. Symp. on Information Theory* (B.N. Petrov, and F. Csaki, eds.), Budapest: Akademiai Kiado, 267-281.

Berk, K.N. (1978). Comparing Subset Regression Procedures, *Technometrics*, **20**, 1-6.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*, Philadelphia: SIAM.

Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *J. Amer. Statist. Ass.*, **78**, 316-331.

Freedman, D. (1983). A Note on Screening Regression Equation, *Amer. Statist.*, **37**, 152-155.

Hjorth, U. (1982). Model Selection and Forward Validation, *Scand. J. Statist.*, **9**, 95-105.

Hocking, R.R. (1976). The Analysis and Selection of Variables in Linear Regression, *Biometrics*, **32**, 1-49.

Kipnis, V., Pinsker, I.Sh. (1983). Simulation Estimators of Losses in Autoregression Prediction (in Russian), VINITI *Monograph* 2404-83, USSR Academy of Science, Moscow.

Kipnis, V. (1987). Model Selection and Predictive Assessment in Multiple Regression, *ASA Proceedings Statist. Comput. Sect.*

Lovell, M. (1983). Data Mining, *Rev. Econ. Statist.*, **LXV**, 1-11.

Mallows, C. (1973). Some Comments on C_p, *Technometrics* **15**, 661-675.

Miller, A. (1984). Selection of Subsets of Regression Variables, *J. Roy. Statist. Soc.*, **A**, **147**, 389-425.

Picard, R., Cook, R. (1984). Cross-Validation of Regression Models, *J. Amer. Statist. Assoc.*, **79**, 575-583.

Pinsker, I.Sh., Trunov, V., Kipnis, V., Aidu, E. (1985). Simulation Estimators for Quality of Decisions (in Russian), *Search for a Relationship and Estimation of the Error* (I.Sh. Pinsker, ed.) Moscow: Nauka, 14-32.

Pinsker, I.Sh., Kipnis, V., Grechanovsky, E. (1985). The Use of the F-Statistic in the Forward Selection Regression Algorithm, *ASA Proceed. Statist. Comput. Sect.*, 419-423.

Pinsker, I.Sh., Kipnis, V., Grechanovsky, E. (1987). The Use of Conditional Cutoffs in a Forward Selection Procedure, *Commun. Statist, - Theor. Meth.*, **A16**, 2227-2241.

Rothman, D. (1968). Letter to the Editor, *Technometrics*, **10**, 432.

POSTERIOR INFLUENCE PLOTS

Robert E. Weiss, University of Minnesota

Abstract

The posterior influence plot, a graphical case influence statistic is introduced. It displays the entire influence of an observation on the posterior distribution of the parameters in a statistical model. The statistic is available for a wide class of models including, but not restricted to, linear, nonlinear, and generalized linear regression.

1. Introduction.

Diagnostics are statistics that aid in the identification of problems with a statistical analysis. Specific examples from linear regression are outlier statistics, leverage values, influence statistics and residual plots. This paper develops a graphical case influence statistic that is available for a wide variety of models.

An influential data point has a large effect on the conclusion of an analysis. In a Bayesian analysis, the conclusion will be the posterior $p(\theta \mid Y)$ of the parameter vector θ given Y, the full data vector. Deleting a single case from the analysis changes the posterior from $p(\theta \mid Y)$ to $p(\theta \mid Y_{(i)})$, where $Y_{(i)}$ denotes the reduced data vector. The problem of influential case analysis is to compare the two densities $p(\theta \mid Y)$ and $p(\theta \mid Y_{(i)})$.

The best solution, if possible, is to plot the two densities on the same graph. This is feasible if θ is only one or two dimensions, but what can be done if θ is high dimensional? This paper finds a one or two dimensional function $\tau_i = \tau_i(\theta, x_i)$ that encompasses all of the influence of y_i on the posterior. The posterior influence plot is a simultaneous display of the posteriors $p(\tau_i \mid Y)$ and $p(\tau_i \mid Y_{(i)})$, one plot for each observation.

2. An alternative Bayesian approach.

Another Bayesian approach to influence analysis is to reduce the comparisons between $p(\theta \mid Y)$ and $p(\theta \mid Y_{(i)})$ to a one number summary, the Kullback divergence between the densities (Johnson and Geisser 1985,1982,1983; Pettit and Smith 1985). The problem with this approach is the interpretation of the resulting numbers. How big is big? Also, several different posterior configurations can produce the same numerical value of the influence statistic. For example, in linear regression, a low leverage high outlier observation can have the same value of an influence statistic as a high leverage observation which is not outlying. Which configuration is the actual cause? The best influence statistic would be a plot of $p(\theta \mid Y)$ and $p(\theta \mid Y_{(i)})$ for all θ in R^K, but this plot is difficult to draw in general. A good graphic should, however, capture much more information than just a single number, so, in this paper we attempt to find a low dimensional function of θ that captures all of the influence of the i^{th} observation on the posterior.

The low dimensional function of θ can be found by inspecting the sampling density $f(y_i \mid \theta, x_i)$ of the i^{th} observation y_i, where x_i denotes the independent variables. In most models, the sampling density depends only on a low dimensional function $\tau_i(\theta, x_i)$ and the observation can influence the posterior of θ only through its influence on the posterior of τ_i.

This function depends on the particular case and on the model. For example, in linear regression, with known variance,

$$\tau_i(\theta, x_i) = x_i^T \theta$$

is just one dimensional. Knowing τ_i determines the sampling density of the i^{th} observation.

In linear regression with unknown variance σ^2,

$$\tau_i(\theta, x_i) = (x_i^T \theta, \sigma)$$

is two dimensional. In nonlinear regression with mean function $E[y_i] = \eta(\theta,x_i)$ and unknown variance, τ_i is

$$\tau_i(\theta,x_i) = (\eta(\theta,x_i), \sigma).$$

For generalized linear models (McCullagh and Nelder 1983) with link function $g(\cdot)$,

$$\tau_i(\theta) = g(x_i^T\theta).$$

These models cover a many of the models in use in statistics.

In section 3 a downdating version of Bayes theorem is presented and also a marginal version of Bayes theorem. The statement that the observation influences $p(\theta \mid Y)$ only through $p(\tau_i \mid \theta)$ is proved. In section 4 the Kullback divergence is introduced as an influence statistic, and a proof is given that the Kullback divergence between the θ posteriors for the i^{th} case depends only on the function $\tau_i(\theta,x_i)$. This provides a second proof that y_i only influences $p(\theta \mid Y)$ through $p(\tau_i \mid Y)$. Finally a nonlinear regression example is given in section 5, followed by discussion in section 6.

3. A downdating version of Bayes theorem.

A downdating form of Bayes theorem is

$$p(\theta \mid Y) = \frac{p(\theta \mid Y_{(i)}) f(y_i \mid \theta,x_i)}{\tilde{f}(y_i \mid Y_{(i)},x_i)}, \qquad (1)$$

where $\tilde{f}(y_i \mid Y_{(i)},x_i)$ is the numerator integrated over the range of θ. Equation (1) can be used in either of two directions, to update the posterior after new data arrives, or to remove an observation for purposes of sensitivity or influence analysis.

Change variables in (1) from θ to (τ_i,ρ_i), where τ_i is the function such that the sampling distribution $f(y_i \mid \theta,x_i)$ is equal to $f(y_i \mid \tau_i(\theta,x_i))$, and ρ_i is chosen to make the transformation one to one. The posterior $p(\theta \mid Y)$ can be written as $p(\tau_i \mid Y)\, p(\rho_i \mid Y,\tau_i)$ and similarly for the reduced data posterior. Then (1) can be rewritten

$$p(\tau_i \mid Y)\, p(\rho_i \mid Y,\tau_i) = \frac{p(\tau_i \mid Y_{(i)}) p(\rho_i \mid Y_{(i)},\tau_i) f(y_i \mid \tau_i)}{\tilde{f}(y_i \mid Y_{(i)},x_i)}. \qquad (2)$$

The extra parameter ρ_i only occurs once on each side of (2) and can be integrated out, giving a marginal Bayes theorem for the τ_i parameter.

$$p(\tau_i \mid Y) = \frac{p(\tau_i \mid Y_{(i)}) f(y_i \mid \tau_i)}{\tilde{f}(y_i \mid Y_{(i)},x_i)}. \qquad (3)$$

Dividing (2) by (3) gives

$$p(\rho_i \mid Y,\tau_i) = p(\rho_i \mid Y_{(i)},\tau_i). \qquad (4)$$

Equation (4) says that the i^{th} observation has no effect on the conditional distribution of ρ_i given τ_i. Thus y_i only has an effect on τ_i, the posterior of ρ_i given τ_i does not depend on whether y_i is in the analysis. Rearranging (1) and (3) gives

$$\frac{p(\theta \mid Y_{(i)})}{p(\theta \mid Y)} = \frac{\tilde{f}(y_i \mid Y_{(i)},x_i)}{f(y_i \mid \tau_i(\theta,x_i))} = \frac{p(\tau_i \mid Y_{(i)})}{p(\tau_i \mid Y)}. \qquad (5)$$

The equality of the outer two ratios will be useful in the analysis of the Kullback influence statistic.

4. The Kullback divergence influence statistic.

The Kullback divergence between the full posterior and the reduced data posterior,

$$K_{(i)}(\theta) \equiv \int \log\frac{p(\theta \mid Y_{(i)})}{p(\theta \mid Y)}\, p(\theta \mid Y_{(i)})\, d\theta, \qquad (6)$$

is a useful generic measure of the influence of the i^{th} observation on the posterior of θ. (See McCulloch (1985) and Bernardo (1985, 1979)). The notation $K_{(i)}(\theta)$ is a short hand notation to say that $K_{(i)}$ measures the influence of the i^{th} case on the posterior of θ. The case statistic $K_{(i)}$ can be conveniently and cheaply computed numerically provided the observations are conditionally independent given the parameter vector.

Equation (6) can be simplified by using equation (5) to substitute for the posterior ratio inside the log, changing variables from θ to (τ_i,ρ_i) and integrating out ρ_i, producing

$$K_{(i)}(\theta) = \int \log\frac{p(\tau_i \mid Y_{(i)})}{p(\tau_i \mid Y)}\, p(\tau_i \mid Y_{(i)})\, d\tau_i \equiv K_{(i)}(\tau_i). \qquad (7)$$

The Kullback divergence between the reduced

data and full data θ posteriors is equal to the Kullback divergence between the reduced data and full data τ_i posteriors!

Equation (7) depends on the two posteriors $p(\tau_i \mid Y_{(i)})$ and $p(\tau_i \mid Y)$, which for a wide variety of models will be densities on R^1 or R^2. The plot of these two densities will exhibit all of the influence of the i^{th} point on θ since the conditional distribution of ρ_i given τ_i does not depend on the outcome of the i^{th} case.

5. An example: Bean Root Cell data.

Table 1 is a list of the data and Figure (1) is a plot of the bean root cell data from Ratkowsky's (1983 p. 88) book on nonlinear regression modeling. The response, y, is water content in 10^{-8} g (Heyes and Brown 1956), plotted against the independent variable x, distance in millimeters from growing tip. A normal theory nonlinear regression model with independent errors was used by Ratkowsky to analyze this data. The mean function is the logistic function

$$E[y_i \mid \theta, x_i] = \frac{\theta_1}{1 + \exp(\theta_2 - \theta_3 x_i)}, \qquad (8A)$$

and the variance is assumed constant and known

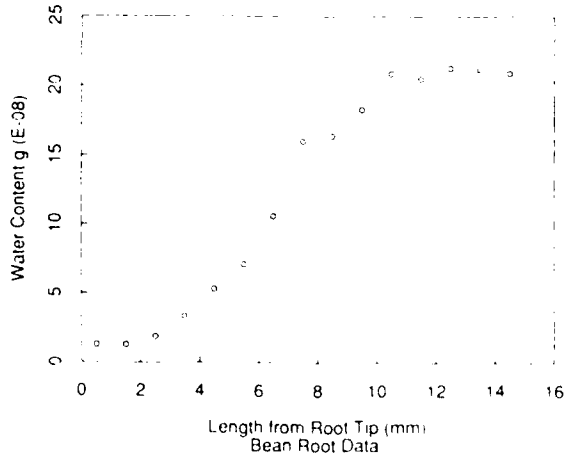
$$Var[y_i] = .414016. \qquad (8B)$$

The Kullback statistics, $K_{(i)}$, were computed for this model using an algorithm based on the iterative Gauss Hermite quadrature methodology of Naylor and Smith (1982).

i	X	Y	$K_{(i)}$	i	X	Y	$K_{(i)}$
1	.5	1.3	.005	9	8.5	16.4	.09
2	1.5	1.3	.002	10	9.5	18.3	.06
3	2.5	1.9	.001	11	10.5	20.9	.18
4	3.5	3.4	.01	12	11.5	20.5	.01
5	4.5	5.3	.01	13	12.5	21.3	.03
6	5.5	7.1	.36	14	13.5	21.2	.01
7	6.5	10.6	.21	15	14.5	20.9	.11
8	7.5	16.0	.98				

Table 1. Bean Root Cell Data and Kullback case influence statistic.

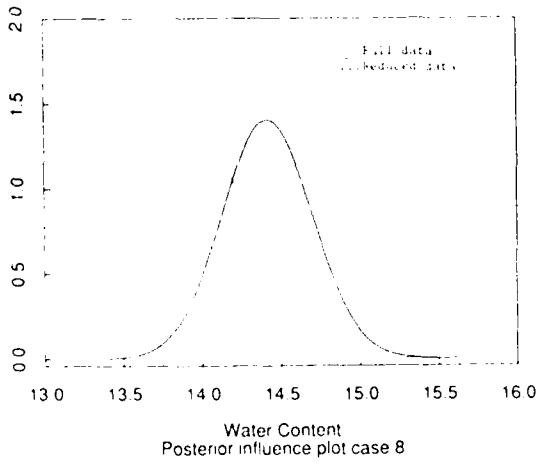
Figure 1.



For the model (8), $\tau_i(\theta, x_i) = \frac{\theta_1}{1 + \exp(\theta_2 - \theta_3 x_i)}$, is a one dimensional parameter and is equal to the sampling mean of the i^{th} observation. That τ_i can be put onto a scale with a physical interpretation is typical of linear, nonlinear and generalized linear models. For this particular nonlinear model, τ_i is length in the same units as the measurements y_i, and consequently the data analyst can use subject matter knowledge to help decide if the observation is having a substantial impact on the inference. Because of the statistical information contained in the plot, the statistician can decide if there are statistical reasons for considering an observation to be highly influential.

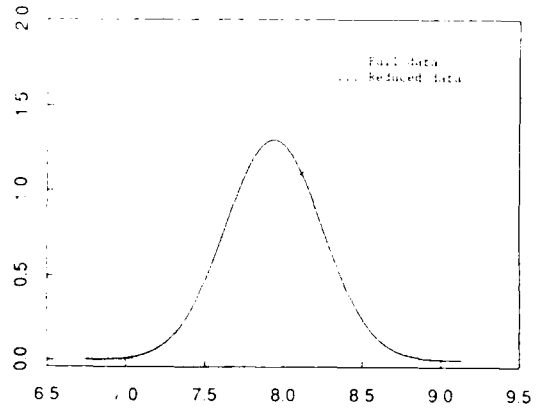
Figures 2 through 5 show four examples of posterior influence plots for the bean root cell data. The influence plots are for observations 8, 6, 14, and 1 respectively. These points have the highest, the second highest, the median, and a very small value of the Kullback influence statistic, respectively. The x-axis is water content in units of $g \times 10^{-8}$ and each plot has the same scale along the x-axis, to facilitate visual comparisons amongst the pictures. Figures 2, 3, and 4 also have the same scale on the y-axis, while figure 5 has a scale 4 times the others to accommodate it's large peak. In each picture, the solid line is the full data posterior marginal $p(\tau_i \mid Y)$ while the

Figure 2.



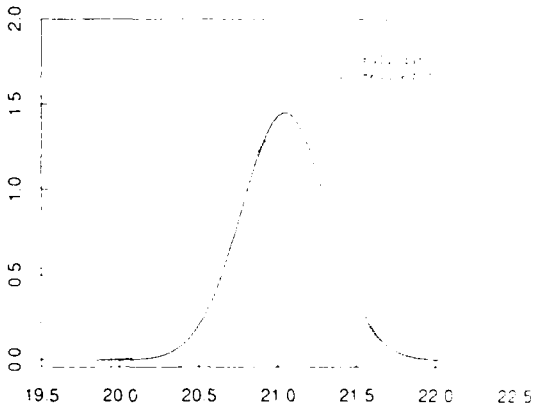
Water Content
Posterior influence plot case 8

Figure 3.



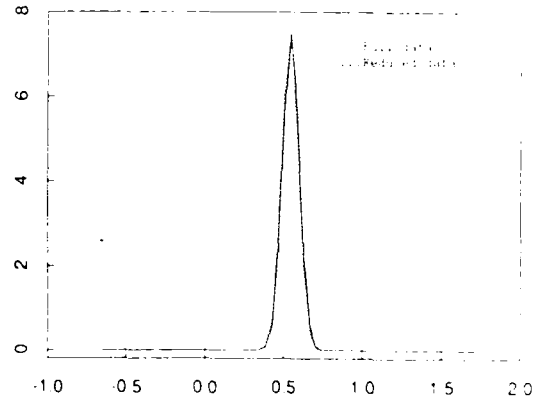
Water Content
Posterior influence plot case 6

Figure 4.



Water Content
Posterior influence plot case 14

Figure 5.



Water Content
Posterior influence plot case 1

dotted line is the reduced data posterior marginal $p(\tau_i \mid Y_{(i)})$.

The most influential observation according to the Kullback statistic is case number 8, with a value of $K_{(i)}$ 2.7 times larger than the next largest value. Figure 2 shows that mode of the posterior decreases by approximately .6 units when the observation is deleted. That the precision in the posterior decreases is also visible because the mode height decreases from approximately 1.4 to 1.2.

Figure 3 shows that omitting the observation number 6 moves probability from lower values of water content to higher values of water content. Deleting the observation also decreases the precision of our posterior.

The next posterior influence plot, figure 4, shows only a moderate influence by observation 14. There is a mild location shift and a mild change in precision.

Figure 5 shows that omitting the first observation has virtually no impact on the posterior. This picture also has a very narrow, sharply peaked density, compared with the other pictures. Given the rest of the data, the location of this observation is quite well determined and consequently it has low influence.

6. Discussion.

There is one posterior influence plot per observation. Rather than drawing every picture, some influence statistic such as $K_{(i)}$ or the L_1 norm between $p(\theta \mid Y)$ and $p(\theta \mid Y_{(i)})$ can be used to select the most interesting pictures. The analyst looks at the posterior

influence plots for observations with the largest values of the selector statistic. If, for example, the plot corresponding to the most influential point does not show a worrisome amount of influence, then no further plots need be looked at.

The posterior influence plot covers all differences between the high-dimensional posteriors $p(\theta \mid Y)$ and $p(\theta \mid Y_{(i)})$, as shown in equation (4). Another interpretation and proof of this statement is as follows.

Define the g-influence of y_i on a function $\beta(\theta)$ as

$$I_g(y_i \text{ on } \beta(\theta)) \equiv \int g\left(\frac{p(\beta(\theta) \mid Y)}{p(\beta(\theta) \mid Y_{(i)})}\right) p(\theta \mid Y_{(i)}) d\theta, \quad (9)$$

where $p(\beta(\theta) \mid Y)$ is the posterior distribution of $\beta(\theta)$ given the data. Then it can be shown that

$$I_g(y_i \text{ on } \theta) = I_g(y_i \text{ on } \tau_i(\theta)) \quad (10)$$

for any measurable g. In a strong sense, the posterior influence plot loses no information about the influence of y_i on the posterior $p(\theta \mid Y_{(i)})$. A statistically interesting example of (9) is the L_1 norm between the full data and reduced data posteriors.

When the parameter $\tau_i(\theta, x_i)$ is two dimensional, then two two-dimensional densities $p(\tau_i \mid Y)$ and $p(\tau_i \mid Y_{(i)})$ need to be compared. Pairs of static contour plots may not be satisfactory, and future work will include the development of a system for looking at pairs of two dimensional densities.

Other work is needed to find good low dimensional projections of $p(\theta \mid Y)$ and $p(\theta \mid Y_{(i)})$ when τ_i is more than two dimensions. Examples where this is interesting are for sets of influential observations, and for multivariate observations.

7. Conclusion.

The posterior influence plot for a particular observation is a graph of the posteriors $p(\tau_i \mid Y)$ and $p(\tau_i \mid Y_{(i)})$. The function τ_i is chosen as the lowest dimensional function of the parameters θ that determines the sampling distribution of the i^{th} observation. The posterior influence plot captures all of the influence of y_i on $p(\theta \mid Y)$, since the marginal densities $p(\rho_i \mid Y, \tau_i) = p(\rho_i \mid Y_{(i)}, \tau_i)$ for any function ρ_i of the parameters. Posterior influence plots are possible in principle for any statistical model, and are practicable for a wide variety of useful statistical models.

Acknowledgements

Thanks to Dennis Cook for many conversations and advice. Thanks to Kathryn Chaloner for proofreading this paper.

REFERENCES.

Bernardo, J.M. (1979). Expected information as expected utility. *Annals of Statistics*, 7, 686-690.

Bernardo, J.M. (1985). Comment on Pettit and Smith's article. In *Bayesian Statistics 2*, J.M. Bernardo *et al.*, eds, p 492-493, Amsterdam: North Holland.

Heyes, J.K. and Brown, R. (1956). Growth and cellular differentiation. In F.L. Milthorpe (Ed.) *The Growth of Leaves*, Butterworth, London.

Johnson, W. and Geisser, S. (1982). Assessing the predictive influence of observations. In *Statistics and Probability Essays in Honor of C.R. Rao*, Kallianpur, Krishnaiah, and Ghosh, eds., Amsterdam: North Holland, 353-358.

Johnson, W. and Geisser, S. (1983). A predictive view of the detection and characterization of influential observations in regression analysis. *Journal of the American Statistical Association*, 78, 137-144.

Johnson, W. and Geisser, S. (1985). Estimative influence measures for the multivariate general linear model. *Journal of Statistical Planning and Inference*, 11, 33-56.

McCullagh, P. and Nelder, J.A. (1983). *Generalized Linear Models*. Chapman and Hall.

McCulloch, R.E. (1985). Model Influence in Bayesian Statistics. Unpublished PhD Dissertation, University of Minnesota.

Pettit, A.N. and Smith, A.F.M. (1985). Outliers and influential observations in linear models. In *Bayesian Statistics 2*, J.M. Bernardo *et al.*, eds, Amsterdam: North Holland, 473-494.

Ratkowsky, D.A. (1983). *Nonlinear Regression Modeling.* Marcel Dekker.

EXACT POWER CALCULATIONS FOR THE CHI-SQUARE TEST OF TWO PROPORTIONS

Carl E. Pierchala, U.S. Department of Agriculture*

ABSTRACT

Approximations are often used when calculating the power of the Pearson Chi-Square test of two independent proportions. This speeds up the computations and simplifies programming. At times, however, it is useful to directly compute the exact power. For example, one may wish to assess an approximation's adequacy in a specific situation. Thus, an APL program was developed to do exact power calculations on an IBM PC/XT. It gives accurate and reasonably fast computations. The exact power values for certain circumstances are compared to the corresponding values obtained using two approximations, one of which is based on the arc sine transformation. It is seen that these approximations are quite inaccurate in some situations.

KEYWORDS: Pearson Chi-Square Test, two-by-two tables, proportions, power, APL, Personal Computers, adverse events, arc sine transformation

1. INTRODUCTION

In a clinical trial being planned to compare a placebo group with an active-treatment group, there was concern that a rare but serious adverse event would be more likely to occur with the active treatment. It was anticipated that the (uncorrected) Pearson Chi-Square test would be used to test the null hypothesis of no difference in proportions of individuals suffering the adverse event. However, the question arose as to whether the study would have sufficient power using this test to detect a several-fold increase in the adverse event rate in the active-treatment group.

Many authorities (e.g., Cohen, 1977; Brownlee, 1965) recommend using the "arc sine" transformation to compute approximate power for a test of equality of two independent proportions. Note that while the test for equality of proportions is often given as a z-test, it follows by an argument analogous to that of Fleiss (1973) that the uncorrected version of the z-test is equivalent to Pearson's test.

Two approximations to the power of a test of equality of two proportions were available in an APL library at the FDA's Center for Drug Evaluation and Research, where the problem motivating this paper first arose. One version (POWHALD), based on the arc sine transformation, was attributed to Hald (1952, pp. 705 ff). The other (POWHSU) was not well documented; it was described as similar to the Hald version, but without an arc sine transformation. Thus, in a preliminary attempt to evaluate the power of the test for the situation at hand, some computations were done using both approximations.

It was assumed that a one-sided, nominal 5% level test would be used, and that both samples would be of size 375. In the placebo group, the probability of the occurence of the adverse event was assumed to be .001. In the treated group, the probability of the adverse event was varied from .001 to .020. Unfortunately, as will be seen in more detail below, the two approximations did not always give very similar values for approximate power. For example, when a treated patient was assumed to have probability .010 of having the adverse event, the Hald approximation gave .59 for the power, while the alternate approximation gave .51. Thus, there was a question as to which, if either, of the two approximations was better.

In addition, it was noted that for an N of 375 and a P of .001, NP equals .375. This is much smaller than 5, which is a conventional criterion for deciding if it is appropriate to do a Chi-Square test (Brownlee, 1965, p. 153). By implication, this led to the question as to whether the arc sine transformation would provide an adequate approximation to the power in such a situation.

To answer these questions, it seemed desirable to attempt to do an exact calculation of the power of Pearson's Chi-Square test. Upon reflection, it became clear that this is conceptually fairly easy to do under a conventional probability model. Development of an APL function to do the computations was thus undertaken. This paper gives a progress report, and reports some computations that shed light on the questions raised above.

2. THEORETICAL BACKGROUND

In this section, the theory behind the computation of the power of the Pearson Chi-Square test is reviewed. The notation used in this paper mimics that used in the code in the APL function that does the computations.

Suppose we observe N_1 identically and independently distributed[1] (IID) bernoulli random variables from one population, and N_2 such variables from a second population.[2] That is,

$X_{11}, X_{12}, \ldots, X_{1N_1}$ are IID $B(1,P_1)$,

and

$X_{21}, X_{22}, \ldots, X_{2N_2}$ are IID $B(1,P_2)$,

with X_{1j} independent of X_{2k}.

---

* The work reported in this paper was begun while the author was employed by the Food and Drug Administration, and was continued after the author moved to the United States Department of Agriculture. The views expressed in this paper are those of the author, and not necessarily those of either the Food and Drug Administration or the United States Department of Agriculture.

The results could be displayed as follows.

POPULATION	SAMPLE SIZE	NUMBER OF SUCCESSES	SAMPLE PROPORTION
1	N_1	S_1	$P_1 = S_1/N_1$
2	N_2	S_2	$P_2 = S_2/N_2$

It follows from the suppositions above that S_1 and S_2 are statistically independent binomial random variates. That is, S_i is $B(N_i,P_i)$, for i=1,2.

Note also that the results could be displayed in a conventional two-by-two table as follows.

	SUCCESSES	FAILURES	TOTALS
POPULATION 1	S_1	N_1-S_1	N_1
POPULATION 2	S_2	N_2-S_2	N_2
TOTALS	S_1+S_2	$N_1+N_2-(S_1+S_2)$	N_1+N_2

Now we may be interested either in a one-sided test,

$$H_0: P_1 \geq P_2 \text{ vs. } H_1: P_1 < P_2,$$

or a two-sided test,

$$H_0: P_1 = P_2 \text{ vs. } H_1: P_1 \neq P_2.$$

In either case we will compute Pearson's Chi-Square statistic,

$$\chi^2 = \frac{(N_1+N_2)[S_1(N_2-S_2)-S_2(N_1-S_1)]^2}{N_1N_2(S_1+S_2)[N_1+N_2-(S_1+S_2)]}.$$

Let α denote the nominal significance level of the test. The one-sided test is significant if both

$$P_1 < P_2 \text{ and } \chi^2 > \chi^2_{1,1-2\alpha}.$$

The two-sided test is significant if

$$\chi^2 > \chi^2_{1,1-\alpha}.$$

Note the difference in critical values.

Now the sample space $\mathcal{S}$ consists of points (S_1,S_2), where $S_1=0,1,2,\ldots,N_1$ and $S_2=0,1,2,\ldots,N_2$. Let R_1 and R_2 denote the rejection regions (i.e., the subsets of points in the sample sapce for which the tests are significant) for the 1-sided and 2-sided tests, respectively.

The probability of any (S_1,S_2) is

$$P_{S_1,S_2} = P_{S_1} \cdot P_{S_2},$$

where

$$P_{S_i} = (N_i!/[S_i!(N_i-S_i)!])P_i^{S_i}(1-P_i)^{N_i-S_i}$$

is a binomial probability, $S_i=0,1,2,\ldots,N_i$, for i=1,2.

Now the power of the one-sided test is given by

$$P_W^{(1)} = \Sigma P_{S_1,S_2}$$

where the sum is taken over (S_1,S_2) in R_1, while the power of the two-sided test is given by

$$P_W^{(2)} = \Sigma P_{S_1,S_2}$$

where the sum is taken over (S_1,S_2) in R_2.

In summary, conceptually it is easy to calculate the power. However in practice there are problems due to the large numer of computations and decisions as to which points in the sample space are in the rejection region.

3. COMPUTATIONAL APPROACH

Now it is to be noted that the sample space is a lattice of points, that is, a rectangular array. Thus for each point in the sample space, the corresponding element of an appropriate matrix can store the probability of each (S_1,S_2), its χ^2 value, and its membership in either R_1 or R_2. Using the APL programming language, one can readily do appropriate matrix computations and form the appropriate sums to obtain the power.

Furthermore, one can speed up the computations by ignoring points in the sample space having trivially small probabilities. Basically, the idea is that there is a rectangular subspace of the sample space "centered" about the expected value (N_1P_1,N_2P_2) of the random variable (S_1,S_2), such that for points outside the subspace the cumulative probability is negligably small. Thus such points can be ignored, so the computations are speeded up by only performing them over the subspace. All that needs to be done is to determine the boundaries of this rectangle, that is, the "marginal subranges" of the S_1 and the S_2 for which the various matrix computations are to be done.

Two different strategies were experimented with for determining these marginal subranges. The approach currently being used is to calculate the P_{S_i}, for $S_i=0,1,\ldots,N_i$, and to only utilize those S_i for which $P_{S_i} > E/N_i$, for some small value, E. A bit of thought shows that the probability of not being in the rectangular subspace is less than 2E, which is thus a conservative upper bound on the error in the computed power.

An earlier strategy which was abandoned was a kind of normal approximation. This involved using S_i in the range $[N_iP_i \pm k(N_iP_i(1-P_i))^{\frac{1}{2}}]$, where k is an appropriate constant. However, although this approach to determining the marginal subranges proved to be computationally quick, it was not always accurate even with k as large as 5

or 6, in particular for small P_1.

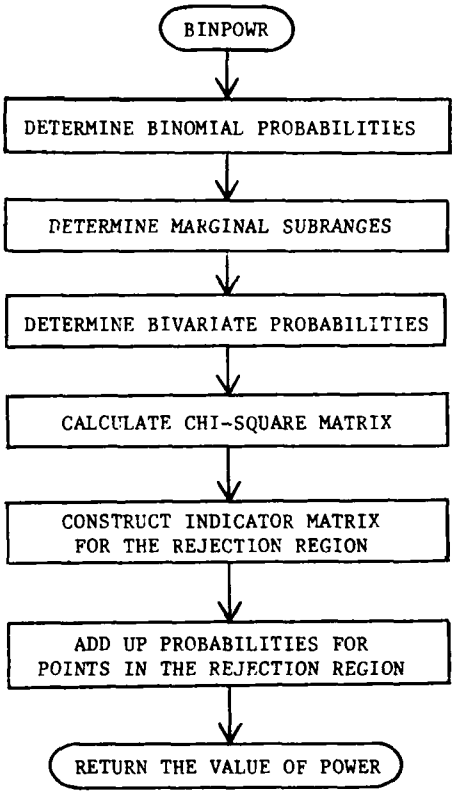
Another technical point is about the computation of the binomial probabilities. This is done in part with a function named LNFAC from the FDA's APL library. This function calculates the base e logarithm of N! (N factorial). Using logarithms and LNFAC, one readily obtains $\log_e P_{S_i}$ and then exponentiates to obtain P_{S_i}.

This indirect approach to calculating the binomial probabilities is slower than the more obvious direct calculation of binomial coefficients multiplied by powers of the appropriate probabilities. However, the APL function that calculates N! won't work for N larger than 170, so when either of the N_1 exceeds 170, the direct approach cannot be used. Using the indirect approach, accurate computations can be done for N_1 larger than 170, making the routine more versatile.

With regard to the calculation of the matrix of chi-square values, which are used in determining the critical region for the test, the computational formula given above is used in conjuntction with APL's matrix capabilities to yield the matrix. The APL code to do so is done in stages in several lines rather than one line, thus reducing the need for temporary storage of intermediate results. This matrix approach proved to be much faster than an earlier approach in which looping was used in combination with a function that computes $\Sigma(o-e)^2/e$ for a four-fold table.

The funtion BINPOWR that does the computations was programmed using Version 6.4 of STSC Inc.'s APL*PLUS on an IBM PC-XT. A general flow chart of the computational proceedure is given in Figure 1.

FIGURE 1. Flow Chart for the Computations.



4. RESULTS

To date, a version of the function BINPOWR has been developed which is accurate and is fairly fast. Some speed results are given in the text below. In developing the routine, extensive checks on the accuracy were made. Power values were calculated for certain simple cases, including the example given by Conover (1971, p. 146). In all such checks, the function gave the correct answer. In addition, Garside and Mack (1976) give exact probabilities of rejecting the null hypothesis of the equality of two proportions. This varies as a function of the common proportion assumed in the computation. Their values could be verified by using BINPOWR with P_1 equal to P_2. In doing so, it was noted that Garside and Mack performed their computations for a one-sided test. At any rate, a variety of Garside and Mack's computations were checked, and in every such case the value computed using BINPOWR agreed with their value to the precision they reported.

However, there are still some problems with the function. First, it is not readily usable as a function. It needs to be made more "user-friendly". Second, the computation of the marginal subranges seems slow. Possibly an approach to speed this up can be found. Third, no estimate is given for the error in the reported exact power value due to restricting the computations to the subspace determined by the marginal subranges. Such an estimate could readily be calculated and returned by the function. Finally, the code for calculating chi-square can be adjusted very simply to include various kinds of continuity corrections. It would be relatively simple and highly desirable to add such a feature to the routine. Thus, BINPOWR is still being refined, and is not yet available for distribution.

The results alluded to in the introduction are given in Table 1. They were computed using an IBM PC-XT with an 8087 math coprocessor chip and 640K memory. With the particular parameters used in this table, approximately 92 to 93 seconds were required for each power value computed using BINPOWR. In the computations, the value $E = 10^{-6}$ was used. Thus, the values in the table are correct to the reported number of digits.

For the set of P_2 values in Table 1, it can be seen that the Hald approximation to the power at least equals and usually exceeds that of the alternate approximation. The difference in nominal power values between the two approximations is greater than .05 when P_2 either is .020 or is in the range from .008 to .012. The difference is greater than .10 when P_2 is in the range from .014 to .018.

TABLE 1. POWER OF THE 1-SIDED, NOMINAL 5% LEVEL PEARSON CHI-SQUARE TEST OF THE EQUALITY OF TWO PROPORTIONS, AS A FUNCTION OF P_2, FOR P_1=.001 AND N_1=N_2=375.

	COMPUTATIONAL METHOD		
P_2	HALD APPROX-IMATION	ALTERNATE APPROX-IMATION	EXACT COMPUTATION
.001	.0500	.0500	.0045
.002*	.0992	.0984	.0278
.004*	.2183	.2056	.1323
.006	.3492	.3140	.2759
.008	.4768	.4165	.4200
.010	.5918	.5098	.5474
.012	.6897	.5925	.6539
.014	.7694	.6643	.7403
.016	.8321	.7258	.8088
.018	.8799	.7776	.8621
.020	.9154	.8209	.9024

* Note change in increment

The exact power values in Table 1 are often substantially different from those given by the approximations. Making comparisons across each row of the table, the following is seen. For small P_2, the exact power values are substantially below the corresponding values from the approximations; this is most striking for P_2 = .001, where the exact value of .0045 is an order of magnitude smaller than that of .0500 given by each approximation. On the other hand, for moderate P_2 (from .006 to .010), the alternate approximation is within .05 of the exact value. For larger P_2 (from .010 to .020), the Hald approximation is within .05 of the exact value. Finally, it can be seen that the alternate approximation gives a value that is higher than the exact power value when P_2 is less than .008; otherwise, the alternative gives a lower value. On the other hand, the Hald approximation always gives a higher value than the exact power value.

5. CONCLUSIONS

A few conclusions can be drawn at this point in the development of BINPOWR. First, STSC's APL on an IBM PC-XT gives accurate, relatively quick exact power computations. This can be useful at the very least for spot checking approximations, which are seen to be inaccurate in some cases. This inaccuracy was particularly extreme in the case where the null hypothesis is true and the common proportion is relatively small. Finally, it is to be noted that APL programming is quite time consuming for the novice APL programmer.

ACKNOWLEDGEMENTS

I wish to thank two individuals at the FDA's Center for Drug Evaluation and Research. Don Schuirmann provided ideas, discussion, and some of the APL functions used by BINPOWR. Bonnie Markowitz provided listings of POWHALD and POWHSU after I had moved on to the USDA.

REFERENCES

1. Brownlee, K. A. (1965), "Statistical Theory and Methodology in Science and Engineering," 2nd Edition, John Wiley & Sons, New York.

2. Cohen, J. (1977), "Statistical Power Analysis for the Behavioral Sciences," Revised Edition, Academic Press, New York.

3. Conover, W. J. (1971), "Practical Nonparametric Statistics," John Wiley & Sons, New York.

4. Fleiss, J. L. (1973), Statistical Methods for Rates and Proportions," John Wiley & Sons, New York.

5. Garside, G. R. and Mack, C. (1976), "Actual Type 1 Error Probabilities for Various Tests in the Homogeneity Case of the 2 x 2 Contingency Table," The American Statistician 30, 18-21.

6. Hald, A. (1952), "Statistical Theory with Engineering Applications," John Wiley & Sons, New York.

On Covariances of Marginally Adjusted Data

James S. Weber, Roosevelt University

A procedure for estimating covariances of marginally adjusted data in terms of first partial derivatives and covariances of the unscaled data and prescribed marginal sums is given. A numerical example demonstrates the dependence of these covariances upon the balancing procedure used to maintain consistency of sets of marginal sums in the presence of errors in the marginal sums.

KEY WORDS. Iterated Proportional fitting algorithm; IPFA; RAS procedure; IPS; Contingency table; Interaction matrix; Gravity model; Input-output model; Partial differentiation; Diagonally equivalent matrices.

1. Introduction

Categorical data may be presented as rectangular tables of rows and columns using two subscripts or as more general arrays with three or more subscripts. Applications of marginally adjusted categorical data include adjusted census data, migration modeling, updating of Leontief input-output coefficients, journey-to-work trip distribution modeling and certain budgeting allocation problems. (See Bacharach, 1970, Weber 1987, etc.)

In this paper we discuss the estimation of covariances of adjusted data in terms of covariances of initial entries and prescribed marginal sums. Two related issues are prominent. 1. The dependence of covariances (and derivatives) on the way that inconsistent marginal sums are made consistent; 2. The calculation of partial derivatives of scaled entries with respect to initial entries and marginal sums. The format of the paper is as follows. In section 2 we specifically describe row and column adjustments of tables of data. In section 3 we look at estimation of covariances of the marginally adjusted data. Then we contrast our approach with others. Also a more general framework is indicated. Finally appendices give a general least squares estimate of marginal sums and a sketch of a proof of convergence of our sequence of derivatives. While some relevant comments have appeared in the literature, few focus explicitly on the manner by which inconsistent constraints are revised to become or remain consistent. (See Weber, 1987). Consistency of marginal sums is central to understanding covariances of marginally adjusted tables or arrays. From time to time we point out details for computing these covariances.

2. Marginally Adjusted Tables of Data

We describe the basic calculating procedure for scaling tables of data to have prescribed row and column sums. A is an $m \times n$ positive matrix of initial values. M^{*1}, M^{*2} are vectors of m row and n column marginal sums. B is an $m \times n$ scaled matrix with the prescribed marginal sums M^{*1}, M^{*2}. D_1, D_2 are diagonal scaling factors. To accommodate variation in marginal sums which at one moment we regard as free to vary in conformity with some covariance matrix but which in some way must also remain consistent, we write M^{*1}, M^{*2} as functions of M^1, M^2, wherein M^1, M^2 vary freely and upon which M^{*1} and M^{*2} depend.

Equations (1abc) (4ab) summarize our setup.

$$(m_i^{*1}) = M^{*1} = \mathcal{F}_1(M^1, M^2) > 0 \tag{1a}$$

$$(m_j^{*2}) = M^{*2} = \mathcal{F}_2(M^1, M^2) > 0 \tag{1b}$$

$$\sum_{i=1}^{m} m_i^{*1} = \sum_{j=1}^{n} m_j^{*2} \tag{1c}$$

$$a_{ij} > 0 \tag{2}$$

$$b_{ij} = d_i^1 a_{ij} d_j^2 \tag{3a}$$

$$B = D_1 A D_2 \tag{3b}$$

$$\sum_{j=1}^{n} b_{ij} = m_i^{*1} \tag{4a}$$

$$\sum_{i=1}^{m} b_{ij} = m_j^{*2} \tag{4b}$$

It is well known that (1c), (3a), (4ab) uniquely determine b_{ij}'s (Sinkhorn, 1967 and many others). We assume that $M^1, M^2 > 0$. Equations (1d), (1e), below, give instances of $\mathcal{F}_1, \mathcal{F}_2$ in (1ab). (See Weber, 1987, p. 626, (5), (6)).

$$m_p^{*1} = m_p^1 + (\sum_{j=1}^{n} m_j^2 - \sum_{i=1}^{m} m_i^1)/(m+n) \tag{1d}$$

$$m_q^{*2} = m_q^2 - (\sum_{j=1}^{n} m_j^2 - \sum_{i=1}^{m} m_i)/(m+n) \tag{1e}$$

Obviously if $\sum_{j=1}^{n} m_j^2 = \sum_{i=1}^{m} m_i^1$ then $m_p^{*1} = m_p^1$ and $m_q^{*2} = m_q^2$.

It is also well known that b_{ij}'s may be calculated by iteratively scaling the initial and subsequent values. This procedure has many names[1] including "Iterated Proportional Fitting Algorithm" or "IPFA". Since this scaling procedure is used to estimate covariances, it is expressed below as (5) followed by the iteration of (6a) and (7a) (or (5) followed by the iteration of (6b) and (7b)).

$$b_{ij}^0 = a_{ij} \tag{5}$$

$$b_{ij}^{2r+1} = \left(\frac{m_i^{*1}}{\sum_{q=1}^{n} b_{iq}^{2r}}\right) b_{ij}^{2r} \qquad r = 0 \ldots \infty \tag{6a}$$

$$b_{ij}^{2r+1} = \left(\frac{m_j^{*2}}{\sum_{p=1}^{m} b_{pj}^{2r}}\right) b_{ij}^{2r} \qquad r = 0 \ldots \infty \tag{6b}$$

$$b_{ij}^{2r+2} = \left(\frac{m_j^{*2}}{\sum_{p=1}^{m} b_{pj}^{2r+1}}\right) \qquad r = 0 \ldots \infty \tag{7a}$$

$$b_{ij}^{2r+2} = \left(\frac{m_i^{*1}}{\sum_{q=1}^{m} b_{iq}^{2r+1}}\right) b_{ij}^{2r+1} \qquad r = 0 \ldots \infty. \tag{7b}$$

That is, (5) followed by iteration of (6a), (7a) (or (6b), (7b)) rapidly converge to a limiting matrix denoted by b_{ij}^{∞} or B^{∞} or simply B. We may write

$$\begin{aligned} B &= B(A, M^1, M^2) \\ &= B^{\infty}(A, \mathcal{F}_1(M^1, M^2), \mathcal{F}_2(M^1, M^2)) \\ &= B^{\infty}(A, M^{*1}, M^{*2}), \text{ etc.} \end{aligned}$$

Often only (5), (6a), (7a) (or (5), (6b), (7b)) are regarded as the well known scaling procedure[1] for computing b_{ij} satisfying (1c), (3a), (4ab). However, for our purposes (1ab) must be regarded as an explicit and integral part of the scaling procedure. Hence our complete description of the marginal adjustment procedure is (1ab), (5), and iteration of (6a) and (7a) (or (6b) and (7b)).

3. Covariances of Marginally Adjusted Tables of Data

We want to estimate the covariance matrix $\mathrm{COV}(B, M^{*1}, M^{*2})$

$$\begin{bmatrix} \mathrm{cov} b_{11} b_{11} & \mathrm{cov} b_{11} b_{12} & \cdots & \mathrm{cov} b_{11} b_{mn} & \mathrm{cov} b_{11} m_1^{*1} & \cdots & \mathrm{cov} b_{11} m_n^{*2} \\ \mathrm{cov} b_{12} b_{11} & \mathrm{cov} b_{12} b_{12} & \cdots & & & & \mathrm{cov} b_{12} m_n^{*2} \\ \vdots & & & & & & \\ \mathrm{cov} m_n^{*2} b_{11} & & \cdots & & & & \mathrm{cov} m_n^{*2} m_n^{*2} \end{bmatrix} \tag{8a}$$

which may be partitioned as

$$\begin{bmatrix} \mathrm{cov}(B,B) & \mathrm{cov}(B,M^{*1}) & \mathrm{cov}(B,M^{*2}) \\ \mathrm{cov}(M^{*1},B) & \mathrm{cov}(M^{*1},M^{*1}) & \mathrm{cov}(M^{*1},M^{*2}) \\ \mathrm{cov}(M^{*2},B) & \mathrm{cov}(M^{*2},M^{*1}) & \mathrm{cov}(M^{*2},M^{*2}) \end{bmatrix} \tag{8b}$$

in terms of $\mathrm{COV}(A,M^1,M^2)$. This can be approximated as an instance of $\mathrm{COV}(F(X)) \cong (\partial F/\partial X)\mathrm{COV}(X)(\partial F/\partial X)^T$, namely

$$\mathrm{COV}(B,M^{*1},M^{*2}) \cong$$

$$\left(\frac{\partial BM^{*1}M^{*2}}{\partial AM^1M^2}\right)\mathrm{COV}(AM^1M^2)\left(\frac{\partial BM^{*1}M^{*2}}{\partial AM^1M^2}\right)^T \tag{9}$$

wherein "(∂/∂)" denotes the appropriate Jacobian matrices of derivatives and T denotes a matrix transpose. We next focus on computing $(\partial BM^{*1}M^{*2}/\partial AM^1M^2)$.

3.2 Derivatives of Scaled Matrices

Some further discussion of notation is required. Expressions (5), (1a), (1b) can be combined to form a single vector function (B^0, M^{*1}, M^{*2}) of (A, M^1, M^2). Obviously $B^0(A)$ simply maps A to the b_{ij} entries via a suitable identity map. M^{*1} and M^{*2} are computed in a single step but are always needed to iteratively scale the matrices B^{2r}, B^{2r+1}. Also (5a) - (6a), (or (5b) - (6b)) can be combined with identity maps for M^{*1}, M^{*2} to define a function $(B^{2r+1}, M^{*1}, M^{*2})$ of (B^{2r}, M^{*1}, M^{*2}). Doing this makes the augmented functions iterate vectors of the same dimension from the beginning to the end of the iteration process allowing the calculus chain rule to apply in a simple way. Finally the augmented scaling (6a) or (7b) may be abbreviated by "R" for a row adjustment augmented by identity maps to carry along M^{*1}, M^{*2}. The augmented (7a) (or (6b)) may be abbreviated by "C" for column adjustment augmented by identity maps to carry along M^{*1}, M^{*2}. That is, $(B^{2r+1}, M^{*1}, M^{*2}) = R(B^{2r}, M^{*1}, M^{*2})$, etc. With these notational changes we may rewrite (9) as

$$\begin{aligned} &\mathrm{COV}(B,M^{*1},M^{*2}) \cong \\ &\left\{\left[\left(\frac{\partial C}{\partial BM^{*1}M^{*2}}\right)\left(\frac{\partial R}{\partial BM^{*1}M^{*2}}\right)\right]^{3\text{ or }4}\left[\frac{\partial B^0M^{*1}M^{*2}}{\partial AM^1M^2}\right]\right\} \\ &\cdot \mathrm{COV}(A,M^1,M^2) \\ &\cdot\left\{\left[\left(\frac{\partial C}{\partial BM^{*1}M^{*2}}\right)\left(\frac{\partial R}{\partial BM^{*1}M^{*2}}\right)\right]^{3\text{ or }4}\left[\frac{\partial B^0M^{*1}M^{*2}}{\partial AM^1M^2}\right]\right\}^T \\ &= \prod_{k=1}^{15\text{ or }19}\Gamma_k \end{aligned} \tag{10}$$

for Γ_k appropriately identified with the above matrices. Note that covariances may be calculated individually since

$$\begin{aligned} \mathrm{cov}(b_{ij},b_{pq}) &= [\mathrm{COV}(B,M^{*1},M^{*2})]_{(i-1)n+j,(p-1)n+q} \\ &= \sum_{k_1=1}^{mn+m+n}\cdots\sum_{k_{18}=1}^{mn+m+n}\gamma^1_{(i-1)n+j,k_1} \\ &\quad\cdot\gamma^2_{k_1k_2}\cdots\gamma^{18}_{k_{17}k_{18}}\gamma^{19}_{k_{18},(p-1)n+q} \end{aligned} \tag{11}$$

wherein $(i-1)n+j$, $(p-1)n+q$ change double indices to a single index.

The preceding gives an overview of a computational procedure. We refer the reader to Weber (1987), Weber and Sen (1985), Weber and Sen (1983) and Weber (1981) for additional details. Here we summarize only key ideas of those papers as they relate to covariances of marginally adjusted data.

Obviously many individual partial derivatives are required to evaluate expressions (10) or (11). Let us now look more closely at these[2].

We may write

$$\left(\frac{\partial R}{\partial B,M^{*1},M^{*2}}\right) = \left(\frac{\partial B^{2r+1},M^{*1},M^{*2}}{\partial B^{2r},M^{*1},M^{*2}}\right) \tag{12a}$$

$$= \begin{bmatrix} \frac{\partial B^{2r+1}}{\partial B^{2r}} & \frac{\partial B^{2r+1}}{\partial M^{*1}} & \frac{\partial B^{2r+1}}{\partial M^{*2}} \\ \frac{\partial M^{*1}}{\partial B^{2r}} & \frac{\partial M^{*1}}{\partial M^{*1}} & \frac{\partial M^{*1}}{\partial M^{*2}} \\ \frac{\partial M^{*2}}{\partial B^{2r}} & \frac{\partial M^{*2}}{\partial M^{*1}} & \frac{\partial M^{*2}}{\partial M^{*2}} \end{bmatrix} \tag{12b}$$

$$= \begin{bmatrix} \frac{\partial b^{2r+1}_{11}}{\partial b^{2r}_{11}} & \frac{\partial b^{2r+1}_{11}}{\partial b^{2r}_{12}}\cdots\frac{\partial b^{2r+1}_{11}}{\partial b^{2r}_{mn}} & \cdots & \frac{\partial b^{2r+1}}{\partial m^{*2}_n} \\ \frac{\partial b^{2r+1}_{12}}{\partial b^{2r}_{11}} & \cdots & & \frac{\partial b^{2r+1}_{12}}{\partial m^{*2}_n} \\ \vdots & & & \vdots \\ \frac{\partial b^{2r+1}_{mn}}{\partial b^{2r}_{11}} & \cdots & & \frac{\partial b_{mn}}{\partial m^{*2}_n} \\ 0 & & I & 0 \\ 0 & \cdots & 0 & I \end{bmatrix} \tag{12c}$$

Obviously the format above bears a family resemblence to the covariance matrices (8ab) for which (12abc) is used to estimate. To put formulae or values in (12c) we may differentiate (6a), (7a) and enter either formulae or values. (See Weber and Sen, 1985a). The entire matrices $\partial R/\partial BM^{*1}M^{*2}, \partial C/\partial B, M^{*1}, M^{*2}$ can be manipulated or formulae for the entries can be coded as functions of indices, B, A, M^{*1}, and M^{*2}. Then covariances (as well as derivatives, if desired) may be computed all at once as a matrix using (10) or as subsets or individually via (11). Obviously there may be a wealth of applicable techniques for fast and efficient handling of large matrices.

3.3 A Numerical Example

The following discussion of a 5×5 example shows that linearly estimated covariances depend on the procedure by which inconsistencies are resolved. Weber (1987) gives a numerical example showing the dependence of constrained gravity model derivatives on the balancing procedure. Mathematically, constrained gravity models are identical to the row and column marginally adjusted tables of data that we have been discussing.

Table 1 gives hypothetical data for A, M^1, M^2 and B. This example is borrowed from Weber (1987; 1981) and is reasonable in the context of those discussions. Here all that matters is that $A, M^1, M^2, B > 0$, which is obviously true.

Table 1
Iterated Proportional Fitting Example

A: Initial Interaction Matrix

0.7828	0.6128	0.4512	0.4331	0.3679
0.6128	0.7363	0.6128	0.5882	0.4996
0.4512	0.6128	0.7515	0.5761	0.4512
0.4331	0.5882	0.5764	0.7214	0.5099
0.3679	0.4996	0.4512	0.5099	0.7068

M^1&M^{*1}: Row Sums

m^1_1	m^1_2	m^1_3	m^1_4	m^1_5
6500.00	8000.00	5280.00	850.00	3730.00

M^2&M^{*2}: Column Sums

m^2_1	m^2_2	m^2_3	m^2_4	m^2_5
6690.00	8030.00	5270.00	840.00	3530.00

B: Marginally Adjusted Values

2431.37	2078.87	1108.14	180.44	701.18
2143.40	2813.23	1694.93	275.98	1072.47
1151.43	1708.13	1516.59	197.29	706.57
189.66	281.36	199.58	42.37	137.03
744.14	1148.42	750.76	143.92	912.76

Note: $M^{*1} = M^1$ and $M^{*2} = M^2$ since $\sum_{i=1}^{m} m_i^1 = \sum_{j=1}^{n} m_j^2$. In B, the row sums are given by M^1, the column sums are given by M^2 and $B = D_1 A D_2$ where D_1 and D_2 are diagonal matrices. B was obtained by the Iterated Proportional Fitting Algorithm.

Table 2, below, gives the partial derivatives, $\partial b_{11}/\partial m_1^1 m_2^1 m_3^1 m_4^1 m_5^1 m_1^2 m_2^2 m_3^2 m_4^2 m_5^2$ for three different balancing procedures: A, B (BL and BR) and C. The procedures A, BL, BR and C are defined immediately after Table 3b. See Weber (1987) for a more lengthy discussion.

TABLE 2
$\partial b_{11}/\partial M^1 M^2$

	m_1^1	m_2^1	m_3^1	m_4^1	m_5^1	m_1^2	m_2^2	m_3^2	m_4^2	m_5^2
A	292	− 0482	− 0330	− 0346	− 0303	283	− 0462	− 0308	− 0322	− 276
BL	343	00248	0177	0161	0204	232	− 0969	− 0815	− 0829	− 0783
BR	243	− 0925	0833	− 0838	− 0795	332	0031	0258	0170	0216
C	323	− 0179	− 0027	− 0043	0	232	− 0969	− 815	− 0829	− 0783

We use these derivatives to estimate covariances for hypothetical covariances of A, M^1, M^2. In the two examples which follow, we let $\mathrm{COV}(A) = 0$. (The effect of including a diagonal covariance matrix $\mathrm{COV}(A)$ simply would be to add a constant to the numerical values we obtain for $\sigma^2_{b_{11}}$ below).

$$\sigma^2_{b_{11}} = \mathrm{cov}(b_{11}, b_{11}) \cong \left(\frac{\partial b_{11}}{\partial A M^1 M^2}\right) \mathrm{COV}(A M^1 M^2) \left(\frac{\partial b_{11}}{\partial A M^1 M^2}\right)^T \tag{13}$$

$$= \left(\frac{\partial b_{11}}{\partial M^1 M^2}\right) \mathrm{COV}(M^1 M^2) \left(\frac{\partial b_{11}}{\partial M^1 M^2}\right)^T, \text{ since } \mathrm{COV}(A) = 0.$$

Example 1. $\mathrm{COV}(A) = 0$, $\mathrm{COV}(M^1 M^2) = \sigma_m^2\ I_{10} =$ a variance times the 10×10 identity matrix. Then

$$\sigma^2_{b_{11}} = \sigma_m^2 (\partial b_{11}/\partial M^1 M^2, \partial b_{11}/\partial M^1 M^2) = \sigma^2 \| \partial b_{11}/\partial M^1 M^2 \|^2$$

wherein $\| \cdot \|$ is ordinary Euclidean length.

TABLE 3A

Balancing Procedure	$\sigma^2_{b_{11}}$
A	$\sigma^2 \cdot 0.17576217$
BL	$\sigma^2 \cdot 0.20150297$
BR	$\sigma^2 \cdot 0.19954164$
C	$\sigma^2 \cdot 0.18753435$

Largest $\sigma^2_{b_{11}} \div$ smallest $\sigma^2_{b_{11}} \cong 1.15$. In some situations, $\sigma_m^2 = 4872 = \overline{m}^{*2}$ would be reasonable

Example 2. $\mathrm{COV}(A) = 0$, $\mathrm{COV}(M^1 M^2) = 10 \times 10$ diagonal matrix of $(0, 0, 0, 1, 1, 0, 1, 0, 1, 0)$. That is, $\sigma^2_{m_4^1} = \sigma^2_{m_5^1} = \sigma^2_{m_3^2} = \sigma^2_{m_4^2} = 1$ and the others are zero.

TABLE 3B

Balancing Procedure	$\sigma^2_{b_{11}}$
A	0.00528653
BL	0.01693739
BR	0.01364130
C	0.01028051

Largest $\sigma^2_{b_{11}} \div$ smallest $\sigma^2_{b_{11}} \cong 3.20$

For both examples, the balancing procedures for M^{*1}, M^{*2} are defined below.

A. M^{*1} and M^{*2} are least squares estimates such that M^{*1}, M^{*2} are consistent and the sum of the squared deviations from M^1 and M^2 is as small as possible. The formula is given by (1d), (1e).

B. $BL\&BR$ correspond to the set of marginal constraints for which the constraints which sum to the larger value are scaled down to sum to the smaller sum. BL and BR are right and left derivatives and probably the variances associated with BR and BL should be averaged.

C. Here, a single marginal sum, m_5^{*1}, absorbs all of the inconsistencies arising from variability of the others.

In this subsection we established our claim that covariances depend on the balancing procedure. Obviously we would like to be able to report having done a simulation as a check on the examples reported here, but this has not been done. (Note however that Weber and Sen, 1985a, compute all of the covariances for a different numerical example by both a linearization and a simulation, obtaining close agreement).

3.4 Computational Alternatives

These covariances may be estimated by using simulations, however there are a number of disadvantages with simulations.

1. Changes in the b_{ij}'s are highly correlated and therefore a large number of simulated points will be necessary. (See Weber and Sen, 1985a, for discussion of an empirical stopping rule which can be used in this situation).

2. All of the b_{ij}'s need to be calculated for each set of simulated values for A, M^1, M^2. In contrast, linearly approximated covariances can be programmed to provide covariances individually, if desired.

3. It is difficult to simulate random vectors with other than a diagonal covariance matrix. This could be a problem for a simulation but not for a linear approximation.

However, for contingency tables which are not too big, when there is access to adequate computing resources, we prefer using both methods over either one by itself. *Note that when a simulation is done, explicit attention to the balancing of inconsistent row and column sums is required!*

Another alternative arises in using an implicit function approach to obtain the partial derivatives that we need. Bacharach (1970) does this. This leads to a generalized inverse of a singular matrix. The choice of a particular g-inverse should be linked to behavioral circumstances. Weber (1987) discusses three balancing procedures in real world settings.

4. More General Situations and Conclusion

Obviously, the linearization of scaling procedures can be done over 3 or more subscripts. Then we would have 3 or more sets of marginal sums, M^1, M^2, M^3. The literature seems to be less extensive on existence results and algorithms for arrays satisfying a certain functional relationship to an initial set of values and having prescribed marginal sums. In debating between linearizations, simulations, balancing procedures and generalized inverses, the same problems apparently remain. If the initial values are nonnegative rather than positive, then the picture is complicated somewhat. The interested reader might consult Feinberg (1983b, 1970) or Berman and Plemmons, (1979).

To conclude, we emphasize again that the error propagation from marginal sums to scaled values depends on the way that inconsistent marginal constraints are brought back into consistency. Error propagation and sensitivity can be looked at via covariance matrices as we have done here or elasticities or derivatives which is done in Weber (1987).

Appendices discuss the least squares consistent marginal sumsfor arbitrary sets of marginal sums as well as the convergence of derivatives of scaled matrices. It would be nice to have general theorems or rules of thumb giving bounds on the variability and dependence of covariances of marginally adjusted data upon the way that marginal sums are kept consistent. This might be a worthwhile future effort. Finally, implementing (11) using the parse trees as suggested by Sawyer (1984) may be especially efficient.

Endnotes

1. The iteration of (6a) (7a) or (6b) (7b) has many names including the Iterative Proportional Fitting Algorithm or "IPFA" (Fienberg and Meyer, 1983a); The Iterative Scaling Procedure or "ISP" (Plackett, 1974, p. 32); The Deming-Stephan-Furness Procedure or "DSF Procedure" after three early users of the procedure - see Sen and Smith; The Furness Procedure; The *RAS* Procedure after the expression giving the functional form of diagonal equivalence; "painting a matrix" to have prescribed row and column sums (George W. Soules). This list probably is not complete.

2. Our treatment of $\partial B/\partial A$ is slighted here since it does not depend on the choice of $(\mathcal{F}_1, \mathcal{F}_2)$, provded that (M^{*1}, M^{*2}) are the same at some evaluation point (A, M^{1}, M^{2}) for different $(\mathcal{F}_1, \mathcal{F}_2)$'s. When (M^{*1}, M^{*2}) is fixed for some value of(M^1, M^2) and A remains fixed, then $\partial A/\partial B$ is not influenced by different choices of $(\mathcal{F}_1, \mathcal{F}_2)$. See Weber (1981), Chapter 6 for full details of $\partial B^0/\partial A$, $\partial B^{2r}/\partial A$, etc.

References & Related Papers

Alonso, W. *Human Settlement Systems*. Cambridge, MA: Ballinger, 1978, Chap. 9.

Bacharach, M. *Biproportional Matrices and Input-Output Change*. Cambridge: Cambridge University Press, 1970.

Berman, A. Plemmons, R. 1979. *Nonnegative Matrices and the Mathematical Sciences*. Academic Press.

Bishop, Y.M.M., S.E. Fienberg, and P.W. Holland. *Discrete Multivariate Analysis: Theory and Practice*. Cambridge, Massachusetts M.I.T. Press, 1975.

Clifford, A.A. *Multivariate Error Analysis*. New York: Wiley, 1973.

Deming, W.E. and F.F. Stephan. "On a Least Squares Adjustments of a Sampled Frequency Table When the Expected Marginal Totals are Known," *Annals of Mathematical Statistics*, 11(1940), 427-444.

Evans, A.W. "Some Properties of Trip Distribution Methods," *Transportation Research*, 4(1970), 19-36.

Fienberg, S.E. and Meyer, M.M. "Log Linear Models and Categorical Data Analysis With Psychometric and Econometric Applications," *Journal of Econometrics*, 22 (1983a), 191-214.

_____. "Iterative Proportional Fitting," *Encylopedia of Statistical Sciences*, Volume 4. New York: Wiley, 1983b, pp. 275-279.

Fienberg, S.E. "An Iterative Procedure for Estimation in Contingency Tables, *Annals of Mathematical Statistics*, 41(1970), 907-917.

Furness, K.P. "Trip Forecasting," unpublished paper presented at a seminar on the use of computers in traffic planning, London, 1962.

Hua, Chang-I and Porell, F. "A Critical Review of the Development of the Gravity Model," *International Regional Science Review*, 4(1979), 97-126.

Lang, S. *Analysis I*, Addison Wesley, 1968.

Naylor, T.H., Balintfy, J.L., Burdick, D.S. and Chu, K., *Computer Simulation Techniques*, John Wiley, New York, 1966.

Plackett, R.L. *The Analysis of Categorical Data*, Griffin's Statistical Monographs, No. 5. London: Charles Griffin, 1974.

Reynolds, H.T. *The Analysis of Cross-Classifications*. New York: Free Press (MacMillan), 1977.

Sawyer, J.W. "First Partial Differentiation by Computer With an Application to Categorical Data Analysis, "The American Statistician, 38(1984), 300-308.

Sen, A.K. and Smith, T. *Modeling Spatial Flows*, unpublished manuscript, Dept. of Urban Planning and Policy, University of Illinois at Chicago.

Sinkhorn, R. "Diagonal Equivalence to Matrices with Prescribed Row and Column Sums," *American Mathematical Monthly*, 74(1967), 402-405.

van Doorn, J. *Disequilibrium Economics*. London: Macmillan, 1975.

Weber, J.S. "Limits of Arbitrary Row and Column Adjustments to Positive Matrices," unpublished manuscript, 1980.

_____. "A Preliminary Supply and Demand Sensitive Respecification of the Gravity Model," Paper presented at the 28th North American Meetings of the Regional Science Association, Montreal, Canada, November 13-15, 1981b

_____. "Derivatives of Constrained Gravity Models," Paper presented at the 30th North American Meetings of the Regional Science Association, Chicago, Illinois, November 11-13, 1983.

_____. "Derivatives of Differentiable Iterative Limits and of D_1AD_2 Matrices," Paper presented at Second SIAM Conference on Applied Lienar Algebra, Raleigh, North Carolina, April 1985.

_____. "Elasticities of Constrained Gravity Models," *Journal of Regional Science*, 27, 1987, 621-640.

Weber, J.S. and Sen, A.K. "On the Sensitivity of Gravity Model Estimates, *Journal of Regional Science*, 25(1985a), 317–336.

_____. "On the Sensitivity of Maximum Likelihood Estimates of Gravity Model Parameters," Springer-Verlag Lecture Notes in Economics and Mathematical Systems, No. 247, 1985b, pp. 148 161.

Wilson, A. G. "A Statistical Theory of Spatial Distribution Models," *Transportation Research*, 1 (1967), 253–270.

_____. "Advances and Problems in Distribution Modelling, *Transportation Research*, 4(1970), 1–18.

_____. "Some New Forms of Spatial Interaction Models: A Review, *Transportation Research*, 9(1975), 167–179.

Appendix 1
Least Squares Consistent Marginal Sums

Given a set of vectors, M^ν, $\nu = 1,\ldots,\nu$, of marginal sums, $M^\nu = (m_i^\nu,\ i = 1,\ldots,I_\nu)$, LaGrange multipliers may be used to obtain revised marginal sums such that

$$\sum_{i=1}^{I_\nu} m_i^{*\nu} = \sum_{i=1}^{I_V} m_i^{*V} \text{ for } \nu = 1,\ldots,(V-1) \tag{A1}$$

$$\sum_{\nu=1}^{V}\sum_{i=1}^{I_\nu}(m_i^{*\nu} - m_i^\nu)^2 \text{ is minimized} \tag{A2}$$

The LaGrangian

$$\mathcal{L}(M,\lambda) = \sum_{\nu=1}^{V}\sum_{i=1}^{I_\nu}(m_i^{*\nu} - m_i^\nu)^2 + \sum_{\nu=1}^{V-1}\lambda_\nu\Big(\sum_{i=1}^{I_V} m_i^{*V} - \sum_{i=1}^{I_\nu} m_i^{*\nu}\Big) \tag{A3}$$

leads to

$$\frac{\partial\mathcal{L}}{\partial m_i^{*\nu}} = 2(m_i^{*\nu} - m_i^\nu) - \lambda_\nu = 0 \quad \text{for } \nu = 1,\ldots,(V-1) \text{ for } i = 1,\ldots,I_\nu \tag{A4}$$

$$\frac{\partial\mathcal{L}}{\partial m_i^{*V}} = 2(m_i^{*V} - m_i^V) + \sum_{\nu=1}^{V}\lambda_\nu = 0 \text{ for } i = 1,\ldots,I_V \tag{A5}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda_\nu} = \sum_{i=1}^{I_V} m_i^{*V} - \sum_{i=1}^{I_\nu} m_i^{*\nu} = 0 \text{ for } \nu = 1,\ldots,(V-1) \tag{A6}$$

Define $m_+^\nu = \sum_{i=1}^{I_\nu} m_i^\nu$ for $\nu = 1,\ldots,V$. Then (A4) - (A6) lead to

$$(\lambda_\nu) = \frac{2}{I_V}\begin{bmatrix} (\frac{I_1}{I_V}+1) & 1 & 1\ldots 1 \\ 1 & (\frac{I_2}{I_V}+1) & 1\ldots 1 \\ \vdots & & \\ 1 & 1 & 1\ldots(\frac{I_{V-1}}{I_V}+1) \end{bmatrix}^{-1}\begin{bmatrix} (m_+^V - m_+^1) \\ (m_+^V - m_+^2) \\ \vdots \\ (m_+^V - m_+^{V-1}) \end{bmatrix} \tag{A7}$$

wherein (λ_ν) denotes the vector of $(V-1)$ LaGrange multipliers

$$m_i^{*\nu} = m_i^\nu + \frac{\lambda\nu}{2} \quad \text{for } \nu = 1\ldots V-1 \text{ for } i = 1\ldots I_\nu \tag{A8}$$

$$m_i^{*V} = m_i^V - \frac{\sum_{\nu=1}^{V-1}\lambda_\nu}{2} \text{ for } i = 1,\ldots,I_V \tag{A9}$$

Setting $V = 2$, $I_1 = m$, $I_2 = n$ in (A7), (A8) and (A9) leads to (1d), (1e) presented in the body of the paper.

Appendix 2
Sketch of Proof of Convergence of Derivatives

The reader may have questioned the convergence of $[\partial C/\partial BM^{*1}M^{*2})(\partial R/\partial BM^{*1}M^{*2})]^n\partial B^0M^{*1}M^{*2}/\partial AM^1M^2$ as n becomes large as well as the relation of this expression to the derivative of the limit of row and column scalings. This question is explored at length in Weber (1981), Chapter 6 and is lesser detail in Weber and Sen (1985ab).

In fact, the sequence of matrix powers that appear in (10) do converge and the limit is the desired Jacobian matrix of derivatives. To prove this we use a basic calculus result on commuting of limits which simply concludes that

$$\lim_{x\to v}\lim_{y\to w} f(x,y) = \lim_{y\to w}\lim_{x\to v} f(x,y) = \lim_{(x,y)\to(v,w)} f(x,y)$$

provided (i) $\lim_{y\to w} f(x,y)$ exists for each x in a relevant set and (ii) $\lim_{x\to v} f(x,y)$ exists uniformly for the set of y vlaues. (eg. see Lang, 1968, p. 134, Theorem 6). This result can apply here by showing (i) that every finite sequence of iterations is differentiable and (ii) that sequences of Newton quotients converge uniformly for some neighborhood of the independent variables. The first requirement is easily met. The matrices and vectors A, M^{*1}, M^{*2} are positive and the iterative scalings are all trapped in a compact positive region of R^{mn+m+n}. As long as A, M^{*1}, M^{*2} remain strictly positive, each scaling is differentiable.

The second requirement seems to require more lengthy discussion. In a compact, sufficiently small ball about (B^0, M^{*1}, M^{*2}), the Newton quotients are a uniformly Cauchy sequence. That is, for all $\epsilon > 0$, there is an integer, $N(\epsilon)$, such that whenever k_1, k_2 are greater than $N(\epsilon)$, then the Newton quotients for k_1, and k_2 iterations of (RC) differ by less than ϵ.

The above facts are precisely the requirements of the theorem. Consequently the limit of a sequence of Newton quotients for IPFA as an increment to a marginal sum becomes arbitrarily small and the limit of using arbitrarily many compositions of derivatives of IPFA scalings commute. In particular, we may let the increments go to zero and proceed to a derivative of the limit of iterative scalings via derivatives of finitely many scalings. Also, the convergence is rapid, hence a 3rd or 4th power is shown in (10).

A cautionary note must be sounded regarding the convergence of compositions of derivatives of iterative scalings. Sinkhorn (1967) and, independently, Weber (1980) showed that

$$L^1 = \lim_{k\to\infty}(RC)^k(A) = (\sum_{i=1}^{m} m_i^1/\sum_{j=1}^{n} m_j^2)\lim_{k\to\infty}(CR)^k(A)$$

$$= (\sum_{i=1}^{m} m_i^1/\sum_{j=1}^{n} m_j^2)L^2 \tag{A10}$$

Where R and C denote row and column scalings and exponents denote repeated application of the function inside the parentheses, etc. Now differentiating an (i,j) entry on each side of (A10) w.r.t. m_1^1 yields

$$\frac{\partial L_{ij}^1}{\partial m_1^1} = \frac{L_{ij}^2}{\sum_{j=1}^{n} m_j^2} + \left(\frac{\sum_{i=1}^{m} m_i^1}{\sum_{j=1}^{n} m_j^2}\right)\frac{\partial L_{ij}^2}{\partial m_1^1} \tag{A11}$$

Setting $m_+^1 = m_+^2$ leads to

$$\frac{\partial L_{ij}^1}{\partial m_1^1} - \frac{\partial L_{ij}^2}{\partial m_1^1} = \frac{L_{ij}^2}{m_+^2} > 0. \tag{A12}$$

Thus any computer implementation of linearizations of scalings should be done with full awareness that certain intermediate calculations do not converge. However, we observe points of accumulation when there are *sets* of marginal constraints, for example, a set of row sums and a set of column sums. A full explanation is found in Weber 1981, 1987.

James S. Weber
P.O.Box 603
Gurnee, Illinois 60013 0603
U.S.A.
(312) 662 5876

Optimizing Linear Functions of Random Variables Having a Joint Multinomial or Multivariate Normal Distribution

J. P. De Los Reyes, University of Akron

1. Introduction. Let $\nu_1,\ldots,\nu_r$ have a joint multinomial distribution with parameters $n,p_1,\ldots,p_r$ ($\Sigma\nu_i=n$, $\Sigma p_i=1$, $p_i>0$). The standardized variables

$$\chi_i=(\nu_i-np_i)/\sqrt{np_i(1-p_i)}\ ,\quad i=1,\ldots,r,$$

have a limiting joint normal distribution with means zero, variances one, and correlation matrix $\mathbf{R}$ of rank r-1:

$$\mathbf{R}=\mathbf{D}-\mathbf{pp}',\quad \mathbf{D}=\mathrm{diag}[1/(1-p_1),\ldots,1/(1-p_r)], \tag{1}$$

$$\mathbf{p}'=\left[\sqrt{p_1/(1-p_1)},\ \ldots\ ,\sqrt{p_r/(1-p_r)}\right].$$

If $p_i=1/r$ for $i=1,\ldots,r$, then $\mathbf{R}$ is *equicorrelated* with common correlation $\rho=-1/(1-r)$.

Suppose numbers $s_1,\ldots,s_r$ are sought that minimize $G(\mathbf{s})$ subject to the probabilistic constraint $\Psi_r(\mathbf{s})\geq 1-\alpha$ $(0<\alpha<1)$. By normal approximation, numbers $x_1,\ldots,x_r$ are then required which minimize $F(\mathbf{x})$ subject to $\Phi_r(\mathbf{x})\geq 1-\alpha$, where we define for constants $a_i>0$,

$$G(\mathbf{s})=\sum_{i=1}^{r}a_is_i,\quad F(\mathbf{x})=\sum_{i=1}^{r}a_ix_i\sqrt{p_i(1-p_i)}\ ,$$

$$\Psi_r(\mathbf{s})=P\{\nu_1\leq s_1,\ \ldots,\ \nu_r\leq s_r\}, \tag{2}$$

$$\Phi_r(\mathbf{x})=P\{\chi_1\leq x_1,\ \ldots,\ \chi_r\leq x_r\},\ \text{and}$$

$$\Phi_r(x,\rho)=P\{\chi_i\leq x,\ \ldots,\ \chi_r\leq x\}$$

in the *symmetric case*, namely, if $\mathbf{R}$ is equicorrelated and $\mathbf{x}=(x,x,\ldots,x)$, *i.e.*, $\mathbf{x}$ also is *equicoordinate*.

If $r=2$, and $a_1=a_2=1$, then binomial probability vectors (s_1,s_2) may be obtained from tables of the cumulative binomial probability distribution [Harvard Univ. Computing Laboratory 1955] by choosing s_1 and s_2 so that the tail probabilities on either side of the binomial distribution of ν_1 are each equal to $\alpha/2$, in effect centering the probability mass $1-\alpha$ (see also Example 1). The direct evaluation of the multinomial sums $\Psi_r(\mathbf{s})$ in (2) involves considerable difficulties if $r\geq 3$, while the corresponding normal probability integral $\Phi_r(\mathbf{x})$ may be evaluated by numerical integration [Milton 1972]. Since vectors $\mathbf{s}$ that minimize $G(\mathbf{s})$ can be obtained from those that minimize $F(\mathbf{x})$ using the formula

$$s_i=np_i+x_i\sqrt{np_i(1-p_i)}$$

we then focus attention on solving the normal case.

For $r\geq 2$, let us define a multivariate analogue of the upper probability point y_α of a distribution, (y_α is the least value such that $P\{\xi\leq y_\alpha\}\geq 1-\alpha$) to be any vector $\mathbf{y}$ for which $P\{\xi_1\leq y_1\ (i=1,\ldots,r)\}\geq 1-\alpha$. These *upper α probability vectors* are generally nonunique since distinct vectors $\mathbf{y}$ can yield the same probabilities. However the vectors $\mathbf{x}$ and $\mathbf{s}$ named above are unique for the specified singular normal distribution and the multinomial distribution, respectively, since they optimize the linear functions F and G of random variables.

2. The optimization problem. In general, probability vectors may be found as follows: Minimize $F=c_1x_1+\ldots+c_rx_r$ $(c_i>0,\ i=1,\ldots,r)$ subject to (a) equality constraints, if any: $FE_i(\mathbf{x})=A_i$ $(i=1,\ldots,m<r)$ and (b) inequality constraints, if any: $FI_i(\mathbf{x})\geq B_i$ $(i=1,\ldots,n)$ where $x_1,\ldots,x_r$ are values taken by r random variables having a joint distribution, and at least one of the constraints involves a probability distribution of the random variables.

The nonlinear programming problem to find optimal upper α probability vectors $\mathbf{x}$ is the following:

$$\text{(3)}\qquad \text{Minimize } F(\mathbf{x})=\sum_{i=1}^{r}a_ix_i\sqrt{p_i(1-p_i)}\ \text{ subject to:}$$

$$\text{(i) } \Phi_r(\mathbf{x})\geq 1-\alpha \text{ and (ii) } \sum_{i=1}^{r}x_i\sqrt{p_i(1-p_i)}>0$$

where (ii) is included in order that the probability function in (i) is nonnegative.

The solution is two-fold: first is the evaluation of $\Phi_r(\mathbf{x})$; second is the optimization part. The optimization routine LPNLP [Pierre & Lowe 1975] and the multidimensional quadrature subroutine MVNORM [Milton 1972, Bohrer & Shervish 1981] provide a complete numerical solution for $1\leq r-1\leq 10$. However, since LPNLP must evaluate the probability integral $\Phi_r(\mathbf{x})$ at numerous points $\mathbf{x}$ to find the optimal probability vector, a good approximation to $\Phi_r(\mathbf{x})$ and other related "computer-ready" formulas that will ease up the computation of $\Phi_r(\mathbf{x})$ are valuable.

Example 1. To illustrate the bivariate case with $a_1=a_2=1$, consider the following "ice cream problem" (first posed by L. Takács at Case Western Reserve University, 1979): At a banquet the dinner menu lists ice creams of two flavors. Independently of the others each of the 1000 guests may order an ice cream of one of the two flavors with probability $\frac{1}{2}$. Which is the smallest number of ice creams of each flavor that must be provided to insure that each guest gets his or her choice with probability $\geq .999$? A solution using normal approximation is given by:

Theorem 1. Let ν_1 have a binomial distribution with parameters n and p_1. Let $\nu_2=n-\nu_1$, $p_2=1-p_1$. Then the numbers s_1,s_2 for which s_1+s_2 is a minimum and such that (i) $s_1+s_2>n$ and (ii) $\Psi_2(\mathbf{s})\geq 1-\alpha$ both hold, are given by $s_1=np_1+x_\alpha\sqrt{np_1(1-p_1)}$ where $x_\alpha=\Phi^{-1}(1-\alpha/2)$, the upper $\alpha/2$ probability point of the standard normal distribution.

Proof: By normal approximation and Lagrange multipliers, $x_1=x_2$, and the conclusion follows.□

The answer to the "ice cream problem" is thus $s=s_1=1000(.5)+3.291[1000(.5)(.5)]^{1/2}$ or about 552 ice creams each (agrees with values obtained using binomial tables), a little more than the expected demand of 500 each but much less than the maximum possible demand of 1000 ice creams of each flavor. In this *single-period* inventory model [Hillier & Lieberman 1967] the additional 104 ice creams are to ward off shortages that may arise, with probability at most .001, from a variability in demand rather than from any delays in delivery or lead time demand since no reorders are made.

In general, for $r=2$ in (3), the optimal pair (x_1,x_2) may be found by using Lagrange multipliers, expressing x_2 as a function of x_1, then applying a one-dimensional search procedure, such as the bisection method [McCormick and Salvadori 1964].

Under a specified multinomial demand in a single-period model, one might consider multi-type products of one kind such as ice creams of r different flavors, concert T-shirts of various colors and sizes, spare parts of a discontinued multicomponent system, airline or train seats to different cities at a given time (compare and contrast with Feller's railroad train seats example [Feller 1968, p.188]), main dishes at an airline's in-flight meal, dated snack items at a coin-operated vending machine, or blood supply at a local blood bank. Such items either become obsolete quickly, spoil easily, are stocked up only once, or have a future that is uncertain beyond a single period. In each of these cases, an upper α

probability vector s would give the smallest supply s_i of an item of type i such that the probability of no shortage is at least $1-\alpha$.

The next three sections deal with various formulations of the integral $\Phi_r(\mathbf{x})$ in an effort to find simple computer-ready formulas to be used in the anticipated numerical integration: section 3 presents $\Phi_r(\mathbf{x})$ as a single iterated integral over the given simplicial domain of integration; section 4 shows that $\Phi_r(\mathbf{x})$ is a sum of its lower-dimensional marginal distribution integrals; while section 5 expresses $\Phi_r(\mathbf{x})$ as a sum of r! iterated integrals of the uncorrelated normal distribution over certain simplices that yield nice limits of integration.

3. Integration over a simplex. An m-dimensional *simplex* is defined to be the intersection in m-dimensional space of m+1 half-spaces such that any m of the bounding hyperplanes of the half-spaces meet in exactly one point, a vertex of the simplex. Also, any m+1 points which do not lie in an m-dimensional space are the vertices of an m-dimensional simplex whose elements are also simplices formed by subsets of the m+1 points, namely, the $\binom{m+1}{1}$ vertices themselves, the $\binom{m+1}{2}$ edges, the $\binom{m+1}{3}$ triangles, the $\binom{m+1}{4}$ tetrahedrons ,..., in general the $\binom{m+1}{k+1}$ cells bounding the simplex which are k-simplices, and finally, the m+1 bounding cells or faces which are m-1 simplices [Coxeter 1963]. When the m(m+1)/2 edges of an m-simplex are all equal, it is called a *regular simplex*.

Example 2. A 3-dimensional simplex (tetrahedron) has 4 vertices, 6 edges, and 4 triangular faces.

Regarding the random variables $\chi_1,\ldots,\chi_{r-1}$ as coordinates of a point χ in r-1 dimensional space E^{r-1} then $\Phi_r(\mathbf{x})$ is the probability that χ falls in the (r-1)-simplex $\mathcal{G}$:

(4) $\mathcal{G}=\{\chi : -\infty \le x_1 \le \chi_1 \le \cdots \le \chi_{r-1} \le -\sum_{i=1}^{r-1}\chi_i\sqrt{p_i(1-p_i)}/\sqrt{p_r(1-p_r)} \le x_r\}$

defined by the system of r inequalities or closed half-spaces of E^{r-1} where the rth inequality is due to the singularity condition on the χ_i's, $\sum\chi_i\sqrt{p_i(1-p_i)}=0$. If G_j denotes the vertex of $\mathcal{G}$ obtained by omitting the jth inequality in (4) and solving the resulting subsystem of linear equations, then G_j has coordinates in terms of the p_i's:

(5) $G_j=(x_1,\ldots,x_{j-1}, x_j^*, x_{j+1},\ldots,x_{r-1}),\ j=1,\ldots,r-1$ where

$$x_j^* = -\sum_{k\neq j} x_k\sqrt{p_k(1-p_k)}/\sqrt{p_j(1-p_j)}\ ;\ G_r=(x_1,\ldots,x_{r-1}).$$

Direct integration over $\mathcal{G}$ yields the following formula for $\Phi_r(\mathbf{x})$, where without loss of generality, $\phi_{r-1}(t)$ is the joint marginal normal density function of $\chi_1,\ldots,\chi_{r-1}$ with covariance matrix $\mathbf{K}$ derived from $\mathbf{R}$ by deleting row r and column r of $\mathbf{R}$.

Theorem 2. For any constants $p_i>0$, and x_i such that $\sum p_i=1$ and $\sum x_i\sqrt{p_i(1-p_i)}<0$, then

(6) $$\Phi_r(\mathbf{x})=P\{\chi\in\mathcal{G}\}=\int_{L_{r-1}}^{U_{r-1}}\cdots\int_{L_1}^{U_1}\phi_{r-1}(t)\,dt_{r-1}\cdots dt_1$$

$$L_k = \frac{-\sum_{i=1}^{k-1}t_i\sqrt{p_i(1-p_i)} - \sum_{j=k+1}^{r-1}x_j\sqrt{p_j(1-p_j)}}{\sqrt{p_k(1-p_k)}}\ ;\ U_k=x_k.$$

Proof: Solve for χ_{r-1} from the last two inequalities in (4) to get $L_{r-1}\le\chi_{r-1}\le x_{r-1}=U_{r-1}$. Solve for χ_{r-2} from the (r-2)nd inequality in (4) and the inequality just obtained to get $L_{r-2}\le\chi_{r-2}\le x_{r-2}=U_{r-2}$, and so on, down to $L_1\le\chi_1\le x_1=U_1$.□

In the symmetric case, the limits in (6) simplify to

(7) $$L_k = -\sum_{i=1}^{k-1}t_i-(r-k)x \quad\text{and}\quad U_k=x,$$

which agree with a formula given earlier by Bland and Owen[1966]. The algorithm MVNORM [Milton 1972] is incorporated into a program MNCDF by this author to evaluate integrals (6), which are then used with LPNLP to solve (3).

4. Integration over infinite rectangles. Let A_i denote the ith inequality defining the simplex $\mathcal{G}$ in (4), so that $\Phi_r(\mathbf{x})=P\{A_1\cap\ldots\cap A_r\}$ is positive if and only if $A_1\cap\ldots\cap A_r\neq\emptyset$ or if $A_1\cup\ldots\cup A_r=E^{r-1}$. By the *inclusion-exclusion method* [Takács 1967, Feller 1968] the probability P_k that exactly k events occur among $A_1,\ldots,A_r$ is:

(8) $$P_k=\sum_{j=k}^{r}(-1)^{j-k}\binom{j}{k}B_j\ ,\ k=0,1,2,\ldots,r\ ,\ \text{where}$$

$$B_k=\sum_{1\le i_1<i_2<\cdots<i_k\le r} P\{A_{i_1}\cap A_{i_2}\cap\cdots\cap A_{i_k}\}.$$

B_k is the kth binomial moment of the number n_r of events occurring among $A_1,\ldots,A_r$; define $B_0=1$.

The normal integral $\Phi_r(\mathbf{x})$ is expressed as a linear combination of its lower-dimensional marginal normal integrals over infinite rectangles as follows:

Theorem 3. Let $\Phi_k(x_{i_1},x_{i_2},\ldots,x_{i_k})$ denote the k-variate nondegenerate joint marginal distribution of $\chi_{i_1},\chi_{i_2},\ldots,\chi_{i_k}$, $1\le k\le r-1$; $\Phi_0(\mathbf{x})=1$. If $\Phi_r(\mathbf{x})>0$, then

(9) $$\Phi_r(\mathbf{x})=\sum_{j=1}^{r}(-1)^{j-1}\sum_{1\le i_1<\cdots<i_{r-j}\le r}\Phi_{r-j}(x_{i_1},\ldots,x_{i_{r-j}})\cdot$$

$$\cdot\ B_{r-1}-B_{r-2}+\cdots\mp 1$$

Proof: $P_0=1-B_1+B_2-\cdots+(-1)^rB_r=1-P\{A_1\cup A_2\cup\ldots\cup A_r\}=0$. The conclusion follows on noting that $\Phi_r(\mathbf{x})=P\{A_1\cap\ldots\cap A_r\}=B_r$ and $P\{A_{i_1}\cap\ldots\cap A_{i_k}\}=\Phi_k(x_{i_1},\ldots,x_{i_k})$.□

In the symmetric case, the $\binom{r}{j}$ terms in the second summation of (9) are identical to $\Phi_{r-j}(x,\rho)$, yielding

(10) $$\Phi_r(x,\rho)=\sum_{j=1}^{r}(-1)^{j-1}\binom{r}{j}\Phi_{r-j}(x,\rho)$$

another formula by Bland and Owen [1966] for the equicorrelated normal distribution. In a way, (6) and (9) extend the earlier formulas (7) and (10), respectively, to the case when $\mathbf{R}$ is not necessarily equicorrelated but satisfy (1).

Bohrer and Shervish [1981] added an inviolable error bound to the algorithm MVNORM when computing the multivariate normal probabilities of rectangular regions only, which this author incorporated into a program MXSK, to evaluate integrals (9) or(10), and used with LPNLP to solve (3).

5. Integration over orthoschemes. A diagonalization and scaling of the covariance matrix $\mathbf{K}$, simplifies the integrand of $\Phi_r(\mathbf{x})$ in (6) but then the limits of integration over the image simplex $\mathcal{H}$ of $\mathcal{G}$ turn out to be complicated. However, by dissecting $\mathcal{H}$ into r! orthoschemes $\mathcal{O}_i$ which are multidimensional analogues of a right triangle [Coxeter 1963], and then exploiting the symmetry of the uncorrelated standardized (*i.e., spherical*) normal density, each integral over an orthoscheme $\mathcal{O}_i$ has nice limits of integration. An *orthoscheme* $\mathcal{O}$ is a k-dimensional simplex such that for some ordering of its vertices, say, $O_1,O_2,\ldots,O_k$, then all the lines O_kO_{k-1}, $O_{k-1}O_{k-2}$,..., O_2O_1, O_1O are mutually perpendicular. In fact each triangle $O_iO_jO_k$ $(i<j<k)$ is right-angled at O_j. If k=3, the tetrahedron is known as *quadrirectangular* since all of its faces are right-

angled. L. Schläfli [1858] first investigated the content of a *hyperspherical simplex*, or simplex constructed on the surface of a hypersphere, through a dissection of the given simplex into spherical orthoschemes. If the vertices of $\mathcal{O}$ are projected radially onto points $P_1, P_1, \ldots, P_{k-1}$ on a unit hypersphere centered at O_k, then $P_1 P_2 \ldots P_{k-1}$ is a *spherical orthoscheme*.

Example 3. For r=3, the integral $\Phi_3(x_1,x_2,x_3)$ equals that of a bivariate normal over a triangle or 2-dimensional simplex $\mathcal{G}$. Upon transforming, $\Phi_3(\mathbf{x})$ is expressible as a sum of bivariate normal integrals over each of 3! right triangles or orthoschemes formed by connecting the origin to each vertex of the image triangle $\mathcal{H}$ and dropping perpendiculars to each side. By symmetry, the integral over each of the six triangles is equal to an integral over a right triangle with vertices (0,0), $(h_1,0)$, (h_1,h_2), which are tabulated as $V(h_1,h_2)$ by the National Bureau of Standards[1959].

If $\mathbf{L}$ is a diagonal matrix with the eigenvalues $\lambda_1,\ldots,\lambda_{r-1}$ of $\mathbf{K}$ on its diagonal, and $\mathbf{P}$ is an orthogonal matrix such that $\mathbf{P}'\mathbf{K}\mathbf{P}=\mathbf{L}$, then the variables $\xi_1,\ldots,\xi_{r-1}$ defined by

(11) $$\xi=\mathbf{L}^{-1/2}\mathbf{P}'\chi$$

are jointly distributed normal with means zero and covariance equal to $\mathbf{I}$, the (r-1)×(r-1) identity matrix; moreover, $P(\chi\in\mathcal{G}) = P(\xi\in\mathcal{H})$. In the symmetric case, if $\mathbf{L}=\mathrm{diag}[1/(r-1), r/(r-1), r/(r-1),\ldots,r/(r-1)]$ [Graybill 1969] and $\mathbf{P}$ equals the transpose of Helmert's original matrix [Lancaster 1965], namely:

(12) $$\mathbf{P}=\begin{bmatrix} 1\sqrt{r-1} & 1\sqrt{2} & 1\sqrt{6} & \cdots & 1\sqrt{(r-1)r} \\ 1\sqrt{r-1} & 1\sqrt{2} & 1\sqrt{6} & \cdots & 1\sqrt{(r-2)(r-1)} \\ 1\sqrt{r-1} & 0 & -2\sqrt{6} & \cdots & 1\sqrt{(r-2)(r-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1\sqrt{r-1} & 0 & 0 & \cdots & -(r-2)\sqrt{(r-1)} \end{bmatrix}$$

then the image $\mathcal{V}$ of $\mathcal{G}$ under the transformation (11) turns out to be a regular simplex $\mathcal{V}$ with center at origin and edge length $e=x\sqrt{2r/(r-1)}$, x given as in (2). The vertex coordinates of $\mathcal{V}$ are precisely the columns V_i of the matrix

(13) $$\mathbf{V}=\mathbf{L}^{-1/2}\mathbf{P}'\mathbf{G}$$

with the vertex coordinates (5) of $\mathcal{G}$ forming the columns G_i of $\mathbf{G}$.

The rotation matrix, corresponding to $\mathbf{P}$ in (11), that will diagonalize an arbitrary nonsingular correlation matrix other than $\mathbf{K}$ is unknown in general. An iterative method, credited to Jacobi [Carnahan *et al.* 1969] transforms a real symmetric matrix into diagonal form by applying a succession of plane rotations.

The (r-1)-dimensional regular simplex $\mathcal{V}^{r-1} = \mathcal{V}$ may be subdivided into r! congruent orthoschemes $\mathcal{J}=J_{(r-1)}J_{(r-2)}\ldots J_1J_0$ where $J_{(r-1)}$ is the center of $\mathcal{V}^{r-1}$; $J_{(r-2)}$ is the center of any one of the (r-2)-dimensional bounding cells, to be denoted by $\mathcal{V}^{r-2}$, of $\mathcal{V}^{r-1}$; J_{r-3} is the center of any one of the (r-3)-dimensional bounding cells, to be denoted by $\mathcal{V}^{r-3}$, of $\mathcal{V}^{r-2}$; ... ; J_1 is the center of any one of the 1-dimensional bounding cells, to be denoted by $\mathcal{V}^1$, of $\mathcal{V}^2$: *i.e.*, J_1 is the midpoint of any one of the edges that bound a face of $\mathcal{V}^{r-1}$; and finally, J_0 is either one of the two endpoints (vertices of $\mathcal{V}^{r-1}$) that bound an edge $\mathcal{V}^1$ of the regular simplex.

Since each edge $\mathcal{V}^1$ is divided into 2 segments, each 2-dimensional cell $\mathcal{V}^2$ or triangular face has 3 such edges, each 3-dimensional cell $\mathcal{V}^3$ or tetrahedron has 4 such triangular faces, ... , each (k-1)-dimensional bounding cell $\mathcal{V}^{k-1}$ has k such (k-2)-dimensional bounding cells, and *finally*, the (r-1)-dimensional simplex $\mathcal{V}^{r-1}$ itself has r such (r-2)-dimensional cells, therefore the total number of ways of joining the center J_{r-1} of $\mathcal{V}^{r-1}$ to a center J_{r-2} of $\mathcal{V}^{r-1}$, to a center J_{r-3} of $\mathcal{V}^{r-2}$, ... , to a center or midpoint J_1 of an edge $\mathcal{V}^1$, to either endpoint J_0 of an edge, is precisely $(r)(r-1)\cdots 3\cdot 2=r!$. All of the r! orthoschemes thus formed are congruent to one another because $\mathcal{V}$ is a regular simplex.

An iterated integration formula for evaluating $\Phi_r(x,\rho)$ (symmetric case) using plane orthoschemes shows that the resulting domain of integration is reduced to just $1/r!$ of the regular simplex because of the symmetry of the uncorrelated normal density:

Theorem 4. Let $\chi_1,\ldots,\chi_r$ have a joint (singular) normal distribution with $E(\chi_i)=0$, $\mathrm{Var}(\chi_i)=1$, $E(\chi_i\chi_j)=-1/(r-1)$. Let $\xi_1,\ldots\xi_{r-1}$ have a joint normal distribution with $E(\xi_i)=0$, $\mathrm{Var}(\xi_i)=1$, $E(\xi_i\xi_j)=0$ and density function $\phi(t_1,\ldots,t_{r-1})$. For any $x > 0$, then

$$\Phi_r(x,\rho) = r!\ \Theta_{r-1}(x) \quad \text{where}$$

(14) $$\Theta_{r-1}(x) = \int_0^x \int_0^{b_1t_1} \int_0^{b_2t_2} \cdots \int_0^{b_{r-2}t_{r-2}} \phi(t)\,dt_{r-1}\,\cdots\,dt_1$$

and $$b_i=\sqrt{(r-i+1)/(r-i-1)} \quad (i=1,\ldots,r-2).$$

Proof: Since $\Phi_r(x,\rho) = P(\chi_i \le x (i=1,\ldots,r)) = P(\chi_i - \chi_{r-1} \le 0 (i=1,\ldots,r-1)) = P(\xi_1,\ldots\xi_{r-1})\in\mathcal{V}) = P(\xi\in\mathcal{H})$, it suffices to show that:

(a) For some orthoscheme $\mathcal{J}^*$ in the simplicial subdivision of $\mathcal{V}$, then $P(\xi\in\mathcal{J}^*)=\Theta_{r-1}(x)$;

(b) If $\mathcal{J}'$ is any one of the r! orthoschemes of $\mathcal{V}$ *distinct from* $\mathcal{J}^*$, then $P(\xi\in\mathcal{J}') = P(\xi\in\mathcal{J}^*)$.

To prove (a) first let $\mathcal{H} = H_{r-1}H_{r-2}\ldots H_1H_0$ be any orthoscheme having vertices

(15) $$H_{r-1}=(0,\ldots,0),\ H_{r-2}=(h_1,0,\ldots,0),\ H_{r-3}=(h_1,h_2,0,\ldots,0),$$
$$\ldots,\ H_1=(h_1,\ldots,h_{r-2},0),\ H_0=(h_1,\ldots,h_{r-2},h_{r-1}),$$

where

$$h_i=x\sqrt{r(r-1)/(r-i)(r-i+1)} \quad (i=1,\ldots,r-1).$$

Thus $P(\xi\in\mathcal{H}) = P(\xi_{r-1}\le 0,\ \xi_{r-1}\le(h_{r-2}/h_{r-1})\xi_{r-2},\ \xi_{r-1}\le(h_{r-2}/h_{r-1})\xi_{r-2},\ \ldots,\ \xi_{r-1}\le(h_2/h_1)\xi_1,\ \xi_{r-1}\le h_1) = \Theta_{r-1}(x)$ since $h_{i+1}/h_i = b_i$ if $i=1,\ldots,r-2$, while $h_1=x$.

Next let $\mathcal{J}^* = J_{r-1}J_{r-2}\ldots J_1J_0$ be the orthoscheme of $\mathcal{V}$ defined in terms of the vertices V_i of $\mathcal{V}$ as follows: J_{r-1} is the center of the simplex formed by $V_1,\ldots,V_r$; J_{r-2} is the center of the simplex formed by $V_1,\ldots,V_{r-1}$; and so on; J_1 is the center of the simplex formed by V_1 and V_2, *i.e.*, J_1 is the *midpoint* of V_1 and V_2; J_0 is the point V_1. Let $\mathbf{H}$ and $\mathbf{J}^*$ be the matrices with columns $H_{r-1},\ldots,H_0$ and $J_{r-1},\ldots,J_0$, respectively. Then

$$P(\xi\in\mathcal{J}^*)=P(\xi\in\mathcal{H})=\Theta_{r-1}(x)$$

since there exists an orthogonal transformation matrix $\mathbf{K}^*$ such that $\mathbf{H}=\mathbf{K}^*\mathbf{J}^*$, namely,

(16) $$\mathbf{K}^*=\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

To prove (b), let $\mathcal{J}'=J'_{r-1}J'_{r-2}\ldots J'_1J'_0$ be any of the r! congruent orthoschemes of $\mathcal{V}$, each having J_{r-1} as a common vertex, $\mathcal{J}'$ distinct from $\mathcal{J}^*$. Let $\mathbf{J}'$ be the matrix with columns $J'_{r-1}, \ldots, J'_0$. There exists an orthogonal transformation matrix, say $\mathbf{T}$, such that $\mathbf{J}^*=\mathbf{T}\mathbf{J}'$, where $\mathbf{T}$ consists of a rotation, or a reflection, or a composition of rotations and reflections. Hence $P(\xi\in\mathcal{J}')=P(\xi\in\mathcal{J}^*)$.□

The following integral recurrence formulas follow directly from Theorem 4 above:

$$(17)\quad \text{(i)}\ \ \Theta_k(x)=\int_0^x \phi(t)\Theta_{k-1}\left(t\sqrt{\tfrac{k+1}{k-1}}\right)dt$$

$$\text{(ii)}\ \ \Phi_r(x,\rho)=r\int_0^x \phi(t)\Phi_{r-1}\left(t\sqrt{\tfrac{r}{r-2}},-\tfrac{1}{r-2}\right)dt$$

Formula (17-ii) was derived earlier by Ruben[1960], Steck and Owen[1962] and John[1966]. Ruben used a "method of sections" where $\mathcal{R}_r$, a regular simplex centered at (0,...,0), is first divided into r simplices $\mathcal{J}_i$ by joining the centroid to the r vertices. The probability content of each $\mathcal{J}_i$ is then obtained by passing (r-1)-flats parallel to the face opposite that vertex of $\mathcal{J}_i$ which coincides with the centroid, and adding up the probability contents over the sections or slabs between parallel flats. Steck and Owen used conditional probabilities and by repeated application of (17-ii) arrived at (14). John indicated the use of probability integrals $V(h_1,...,h_k)$ over special simplices he called "k+1-hedrons"(which are orthoschemes with vertices as in (15)), to evaluate the probability of convex polyhedra in k-dimensional space under normal and t distributions, deriving (17-ii) for his "k+1-hedrons".

If $u_r=\xi_r-\bar{\xi}$, the extreme order statistic minus the mean in normal samples of r observations, with $E(\xi_i)=0$, $Var(\xi_i)=1$, then $\Phi_r(x,\rho)=P\{u_r\le x\sqrt{(r-1)/r}\}$. A recurrence formula for u_r [Grubbs 1950, David 1970] used in computing its percentage points may be obtained independently using the integral recurrence formula (17-ii) above.

Example 4. Consider the symmetric case for r=6, $p_1=\cdots=p_6=1/6$ so that $\rho=-1/5$, and let $x_1=\cdots=x_6=1.5$. Number in boxes denote [CPU] on the VAX/VMS 3.1. From (10) and program IXSK,

$$\Phi_6(1.5,-1/5)=B_6-B_5+B_4-B_3+B_2-B_1=1$$
$$=\tbinom{6}{1}\Phi_5(1.5,-1/5)-\tbinom{6}{2}\Phi_4(1.5,-1/5)+\tbinom{6}{3}\Phi_3(1.5,-1/5)$$
$$-\tbinom{6}{4}\Phi_2(1.5,-1/5)+\tbinom{6}{5}\Phi_1(1.5,-1/5)-1$$

where

$\Phi_5(1.5,-1/5)=.6841 \pm .0001$ [2558.6]
$\Phi_4(1.5,-1/5)=.7437\ 013 \pm .0000\ 001$ [241.8]
$\Phi_3(1.5,-1/5)=.8050\ 531 \pm .0000\ 001$ [2.5]
$\Phi_2(1.5,-1/5)=.8682\ 136 \pm .0000\ 001$ [0.2]
$\Phi_1(1.5,-1/5)=.9331\ 926 \pm .0000\ 001$ [0.1]

Hence $\Phi_6(1.5,-1/5)=.626 \pm .001$. [2803.2]

From (14) and program MNCDF,

$\Phi_6(1.5,-1/5)=6!\cdot(.0008\ 6998 \pm .0000\ 0001)$ [539.8]

Hence $\Phi_6(1.5,-1/5)=.626\ 38 \pm .000\ 01$.

Earlier attempts to compute $\Phi_r(1.5,-1/5)$ using formulas (6) and (7) with MNCDF have been unsuccessful due either to nonconvergence or to this author's tendency to abort the run whenever it was taking "much too long".

6. Approximations to $\Phi_r(x)$. By the *Bonferroni inequality* [Kotz and Johnson 1982],

$\Phi_r(x) = P\{A_1\cap\cdots\cap A_r\} \ge 1-\sum_{i=1}^{r}P\{A_i^c\} = \sum_{i=1}^{r}P\{A_i\}-(r-1)$, so

$$(18)\qquad \Phi_r(x) \ge \sum_{i=1}^{r}\Phi(x_i)-(r-1)$$

where A_i^c denotes the complement of A_i. If r=2, (18) gives the exact value of $\Phi_r(x)$.

Let S_k denote the kth binomial moment of the number of events m_r occuring among $A_1^c,\ldots,A_r^c$. Then $\Phi_r(x) = P\{A_1\cap A_2\cap\cdots\cap A_r\} = P_0$, the probability that exactly none of $A_1^c,\ldots,A_r^c$ occurs. From (8), $P_0=\sum_{t=0}^{r}(-1)^tS_t$, where $S_0=1$ and $S_r=P\{A_1^c\cap\cdots\cap A_r^c\} = 0$ imply that alternatively, the normal integral equals

$$(19)\qquad \Phi_r(x)= 1-S_1+S_2-\cdots\pm(-1)^{r-1}S_{r-1},$$

where $S_k=\sum_{1\le i_1<\cdots<i_k\le r}P\{A_{i_1}^c\cap\cdots\cap A_{i_k}^c\}$, $k=1,\ldots,r-1$,

yielding a finite sequence of approximations,

$$(20)\qquad \Phi_r^k(x) = 1-S_1+S_2-\cdots\pm(-1)^kS_k,\quad k=1,\ldots,r-1,$$

with (18) as first approximation when k=1, and having error bounds S_2, the first neglected term:

$$(21)\qquad S_2=\sum_{1\le i_1<i_2\le r}P\{A_{i_1}^c\cap A_{i_2}^c\}=\sum_{1\le i_1<i_2\le r}P\{x_{i_1}\le x_{i_1},\ x_{i_2}\le x_{i_2}\},$$
$$\text{or}\quad S_2=\binom{r}{2}\cdot P\{x_1\ge x,\ i=1,2;\ \rho=-1/(r-1)\}$$

in the symmetric case. The sum of an odd number of terms provides an upper bound and the sum of an even number a lower bound, counting the first term, 1. The bounds increase in sharpness with the number of terms included and the magnitude of the error ϵ_k in the kth approximation does not exceed the first neglected term. The following summarizes results on these bounds:

Theorem 5. (i) Bonferroni inequalities: For $r\ge 2$, and $m=1,\ldots,r/2$,

$$\sum_{t=0}^{2m-1}(-1)^tS_t \le \Phi_r(x) \le \sum_{t=0}^{2m}(-1)^tS_t$$

(ii) Improved Bonferroni inequalities: For $r\ge 2$, $s\ge 0$,

$$\sum_{t=0}^{2s+1}(-1)^tS_t+\left(\tfrac{2s+2}{r}\right)S_{2s+2} \le \Phi_r(x)$$
$$\le \sum_{t=0}^{2s}(-1)^tS_t-\left(\tfrac{2s+1}{r}\right)S_{2s+1}$$

(iii) "Best upper bound" using only S_1 and S_2: Let [x] denote the greatest integer in x.

$$\Phi_r(x) \le 1-\frac{2}{k+1}S_1+\frac{2}{k(k+1)}S_2,\quad k=1+\left[\frac{2S_2}{S_1}\right]$$

Proof: (i) See Feller[1968], Kotz & Johnson[1982], David[1970], Galambos[1975]. (ii) See Sobel & Uppuluri [1972], Galambos[1975]. (iii) See Dawson & Sankoff [1967], Kounias & Marin[1976], Galambos[1978].

Table 1 below shows how good an approximation (18) is, considering it involves only the univariate normal distribution, based on computed values of the error bounds (21) using programs MNCDF and IXSK. The values of x were grouped together according as $.5\cdot 10^{-t-1}\le S_2(x)\le .5\cdot 10^{-t}$.

Table 1

Values of $x=x(r,t)$ such that for $x\ge x(r,t)$ the function $\Phi_r^1(x)=1-S_1$ approximates the true value of $\Phi_r(x,\rho)$ to t or more correct decimals, $1\le t\le 4$ and $3\le r\le 30$.

t \ r=	3	4	5	6	7	8	9
1	0.70	1.05	1.25	1.40	1.50	1.60	1.65
2	1.15	1.50	1.70	1.85	1.95	2.05	2.05
3	1.55	1.90	2.10	2.25	2.35	2.45	2.50
4	1.85	2.25	2.45	2.60	2.70	2.80	2.85

t \ r=	10	11	12	13	14	15	16
1	1.75	1.80	1.85	1.90	1.90	1.95	2.00
2	2.20	2.25	2.30	2.30	2.35	2.40	2.40
3	2.55	2.60	2.65	2.70	2.75	2.75	2.80
4	2.90	2.95	3.00	3.05	3.05	3.10	3.10

t \ r=	17	18	19	20	21	22	23
1	2.00	2.05	2.05	2.10	2.10	2.15	2.15
2	2.45	2.45	2.50	2.50	2.55	2.55	2.55
3	2.80	2.85	2.85	2.90	2.90	2.90	2.95
4	3.15	3.15	3.20	3.20	3.25	3.25	3.25

t \ r=	24	25	26	27	28	29	30
1	2.20	2.20	2.20	2.25	2.25	2.25	2.30
2	2.60	2.60	2.60	2.65	2.65	2.65	2.70
3	2.95	2.95	3.00	3.00	3.00	3.00	3.05
4	3.25	3.30	3.30	3.30	3.30	3.30	3.35

Table 2 gives the corresponding values of $\Phi_r(x,\rho)$

at x=x(r,t) as given in Table 1. It is often the case that in (3), α is taken equal to 0.20 or less, and Table 2 indicates for example, that if $1-\alpha \geq .90$, the approximation yields 4 correct decimals if r=3, and only 2 correct decimals if r=30. As the dimension r increases, fewer correct decimals are obtained for the same α-value. Note however that for $1-\alpha \geq .99$, (18) gives 4 correct decimals for r=3,....30.

Table 2
Values of $\Phi_r^{(1)}(x)=1-S_1 \approx \Phi_r(x,\rho)$ at x=x(r,t) where x(r,t) is given in Table 1

t \ r=	3	4	5	6	7	8	9
1	.2741	.4126	.4718	.5155	.5324	.5616	.5548
2	.6248	.7328	.7772	.8071	.8209	.8385	.8392
3	.8183	.8851	.9107	.9267	.9343	.9429	.9441
4	.9035	.9511	.9643	.9720	.9757	.9796	.9803
t \ r=	10	11	12	13	14	15	16
1	.5994	.6048	.6141	.6267	.5980	.6162	.6360
2	.8610	.8655	.8713	.8606	.8686	.8770	.8688
3	.9461	.9487	.9517	.9549	.9583	.9553	.9591
4	.9813	.9825	.9838	.9851	.9840	.9855	.9845
t \ r=	17	18	19	20	21	22	23
1	.6132	.6367	.6165	.6427	.6248	.6529	.6371
2	.8786	.8714	.8820	.8758	.8869	.8815	.8761
3	.9566	.9607	.9585	.9627	.9608	.9590	.9635
4	.9861	.9853	.9869	.9863	.9879	.9873	.9867
t \ r=	24	25	26	27	28	29	30
1	.6663	.6524	.6385	.6699	.6577	.6455	.6783
2	.8881	.8835	.8788	.8913	.8873	.8833	.8960
3	.9619	.9603	.9649	.9636	.9622	.9609	.9657
4	.9862	.9879	.9874	.9869	.9865	.9860	.9879

Example 5. Continuing Example 4, formula (19) and IXSK together yield in the symmetric case,

$$\Phi_6(1.5,-1/5)=1-S_1+S_2-S_3+S_4-S_5$$
$$=1-\binom{6}{1}\Phi_1^c(1.5,-1/5)+\binom{6}{2}\Phi_2^c(1.5,-1/5)-$$
$$\binom{6}{3}\Phi_3^c(1.5,-1/5)+\binom{6}{4}\Phi_4^c(1.5,-1/5)-\binom{6}{5}\Phi_5^c(1.5,-1/5).$$

where we define

$\Phi_k^c(1.5,-1/5)=P(A_1^c \cap \ldots \cap A_k^c)$ so that

$\Phi_1^c(1.5,-1/5)=.0668\ 069 \pm .0000\ 001$ [0.01]
$\Phi_2^c(1.5,-1/5)=.0018\ 282 \pm .0000\ 001$ [0.18]
$\Phi_3^c(1.5,-1/5)=.0000\ 098 \pm .0000\ 001$ [1.48]
$\Phi_4^c(1.5,-1/5)=.0000\ 00 \pm .0000\ 01$ [28.57]
$\Phi_5^c(1.5,-1/5)=.000 \pm .001$ [94.17]

Hence $\Phi_6(1.5,-1/5) = .62638 \pm .001$ [124.41]

From (18), $\Phi_6(1.5,-1/5) \approx 1-S_1=6 \cdot \Phi_1(x) - 5$
$\approx .5991\ 586 \pm .0000\ 01$

The error of this appoximation does not exceed S_2, and from (21), $S_2=.0274\ 23 \pm .0000\ 01$.
Hence $\Phi_6(1.5,-1/5) = .5991\ 586 \pm .0274\ 23$
From Table 1, for r=6, x=1.5, *i.e.*, $x \geq 1.40$, the number t of correct decimals is 1.

Since the integrals in these formulas for $\Phi_r(x)$ must be computed numerically (unless otherwise known), it is possible for an approximation using (20) to have an actual error greater than the first neglected term because of errors in the numerical quadrature. It is wise to plan to compute enough decimal digits, in anticipation of the additivity of the error bounds under linear combinations, so as not to render the results meaningless. On the other hand, there is no need to compute numerically the integral terms much more accurately than the specified error bounds.

7. Solution of the total optimization problem.

Up till now the emphasis has been on evaluating the singular normal integral $\Phi_r(x)$, with four formulas (6), (9), (14) and (19) ready for computer implementation. Note that while both B_k's in (9) and S_k's in (19) are sums of lower-dimensional integrals over infinite rectangles, the S_k's in fact yield "upper tail probabilities" of these marginal normal distributions and therefore decline rapidly, (see Examples 4 and 5) yielding approximation (18).

To put the problem in a form suitable for LPNLP, rewrite (3) equivalently as a maximization problem:

$$\text{Maximize } F(\mathbf{x}) = -\sum_{i=1}^{r} a_i x_i \sqrt{p_i(1-p_i)} \text{ subject to:}$$
$$(22) \quad (i) -\Phi_r(\mathbf{x}) \leq -(1-\alpha) \text{ and } (ii) -\sum_{i=1}^{r} x_i \sqrt{p_i(1-p_i)} \leq 0.$$

With (18), the gradient of the probability function in (22-i) has only univariate normal density functions for its components. Since $\Phi(x)$ is a concave function over $x \geq 0$, the set S of feasible solutions is a closed convex set. Therefore the absolute maximum of F over S, F being linear, is the only local maximum over S [Pierre and Lowe 1975].

Program OPRVEC implements LPNLP on the VAX to solve (22), with calls made to a suitable version of MNCDF whenever $\Phi_r(x)$ is to be evaluated. Copies of OPRVEC are available at cost by writing to J. P. de los Reyes, Department of Mathematical Sciences, Univ. of Akron, Akron, Ohio 44325; tel:(216) 375-7193.

Simulation is an alternative approach to solving (22) or similar problems involving other probability distributions such as Poisson. On the other hand, the methods of parallel programming might speed up the numerical quadrature portion of the present solution in which, for instance in the nonsymmetric case of (14), the integral $\Theta_{r-1}(x)$ must be evaluated r! times but with different upper limits.

Example 6. Consider finding an optimal inventory policy for a local hospital's blood bank, in the sense that the daily supply s_i of type i blood is a minimum under a preset probability constraint that no shortage occurs that day with probability at least $1-\alpha$. Suppose that the daily demand ν_i of human blood of type i (i=1,...,r) has joint multinomial probability distribution with parameters n, $p_1,\ldots,p_r$, where p_i is the probability that the bank receives an order of *one unit of type i blood independently of other* requisitions and there are n orders received daily.

First, estimates of the numbers p_i, based on actual percentages used (2369 pints total) over 30 randomly selected days within January to October of one year were found to be:

i=	1	2	3	4	5	6	7	8
type:	O+	A+	B+	A−	O−	AB+	B−	AB−
p_i=	.387	.309	.117	.079	.052	.032	.022	.002

Next, there being no other constraints other than supply levels s_i having a least total, we set a_i's 1 in function G and consequently, a_i's=1 in F also. Now solve the normal case (22) for x, from the given values of a_i, p_i, r, and desired values of α. Use OPRVEC, with approximation (18) for quick results, to obtain Table 3. Equicoordinate vectors (x,...,x) for the symmetric case are in the last row of Table 3.

Finally, the corresponding optimal multinomial vectors $(s_1,\ldots,s_r)$ representing the minimum daily blood supply levels, for the desired risk levels, assuming n=1000, are summarized in Table 4. The first column of Table 4 gives the expected demands based on the estimates of p_i above. Recall the assumption that no resupply can occur in the same *day, hence note the overstocking that increases* as the probability $1-\alpha$ of no shortage increases. If we assume that $p_1=p_2=\ldots=p_8=1/8$, then the average demand is 125 at n=1000. The optimal supply levels s

and totals are shown in the last two rows of Table 4.

Table 3
Normal Probability Vectors for Blood Bank
(see Example 6)

x\1-α =	.90	.95	.99	.995	.999
x_1 =	1.9686	2.2538	2.8217	3.0391	3.4853
x_2 =	1.9953	2.2771	2.8404	3.0565	3.5004
x_3 =	2.1689	2.4307	2.9649	3.1726	3.6022
x_4 =	2.2491	2.5024	3.0240	3.2279	3.6510
x_5 =	2.3357	2.5806	3.0890	3.2888	3.7050
x_6 =	2.4282	2.6646	3.1595	3.3551	3.7640
x_7 =	2.5016	2.7316	3.2163	3.4086	3.8118
x_8 =	2.9333	3.1318	3.5624	3.7370	4.1081
x =	2.2414	2.4977	3.0233	3.2269	3.6357

Table 4
Multinomial Probability Vectors for Blood Bank
(see Example 6)

$E(\nu_1)$\s_1 \1-α=	.90	.95	.99	.995	.999
387 = s_1 =	417	421	430	434	440
309 = s_2 =	338	342	350	353	360
117 = s_3 =	139	142	147	150	154
79 = s_4 =	98	100	105	106	110
52 = s_5 =	68	70	73	74	77
32 = s_6 =	46	47	50	51	54
22 = s_7 =	34	35	37	38	40
2 = s_8 =	6	7	7	8	8
1000 /total/	1146	1164	1199	1214	1273
125 = s_9 =	148	151	157	159	163
1000/total/	1184	1208	1256	1272	1304

For comparison with other values already published, OPRVEC, when run with (18), obtained upper α=.10 and .005 probability vectors in the symmetric case, which agreed to 2 and 3 decimal places respectively, to those obtained through percentage points of the order statistic u_r mentioned after (17) above. When a_i's=1 in the symmetric case, there is really no need to use OPRVEC on (22) since it is plausible that the required vector x must be equicoordinate, thus $x_1=\Phi^{-1}(1-\alpha/r)$, the upper α/r probability point of the standard normal distribution. OPRVEC of course produces the same equicoordinate vectors x in this case.

Practical applications of probability constrained or "chance-constrained" programming in general include the models of minimum cattle feed [Bracken and McCormick 1968] and hog feed rations [Pierre and Lowe 1975] under probabilistic protein constraints, an optimal cost nutrition program under probabilistic nutrient level constraints [Prekopa 1970], and an optimal spare parts kit for a multicomponent system in which demand for spares is generated by component failures having an exponential distribution [Proschan 1960].

REFERENCES

Bland, R. P. and Owen, D. B. (1966). A note on singular normal distributions. *Ann. Inst. Statis. Math. ,Tokyo, 18:113-116.*

Bohrer, R. and Shervish, M. J. (1981). An error-bounded algorithm for normal probabilities of rectangular regions. *Technometrics, 23(3):297-300.*

Bracken, J. P. and McCormick, G. P. (1968). *Selected Applications of Nonlinear Programming.* New York: Wiley.

Coxeter, H. S. M. (1963). *Regular Polytopes, 2nd. ed.,* New York: Mcmillan Co.

David, H. A. (1970). *Order Statistics.* New York: Wiley.

Dawson, D. A. and Sankoff, D. (1967). An inequality for probabilities. *Proc. Amer. Math. Soc., 18:504-507.*

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications, 3rd. ed., Vol.1.* New York: Wiley

Galambos, J. (1975). Methods for proving Bonferroni type inequalities. *J. London Math. Soc. (2), 9:561-564.*

_____. (1978). *The Asymptotic Theory of Extreme Order Statistics.* New York: Wiley.

Graybill, F. A. (1969). *Introduction to Matrices With Applications in Statistics.* Belmont, California: Wadsworth.

Grubbs, F. E. (1950). Sample criteria for testing outlying observations. *Ann.Math.Statist.,21:27-58.*

Harvard University Computing Laboratory (1955). *Tables of the Cumulative Binomial Probability Distribution.*Cambridge,Mass.:Harvard Univ. Press.

Hillier, F. S. and Lieberman, G. J. (1967). *Operations Research, 2nd. ed.,* San Francisco: Holden-Day, Inc.

John, S. (1966). On the evaluation of probabilities of convex polyhedra under multivariate normal and t-distributions. *J. Roy. Statist. Soc. Ser.B, 28:366-369.*

Kotz, S. and Johnson, N. L., editors-in-chief. (1982). *Encyclopedia of Statistical Sciences, Vol. 1.* New York: Wiley.

Kounias, S. and Marin, J. (1976). Best Linear Bonferroni Bounds. *Siam J. Appl. Math., 30:307-323.*

Lancaster, H. O. (1965). The Helmert Matrices. *American Math. Monthly, 72:4-11.*

McCormick, M. J. and Salvadori, M. G. (1964). *Numerical Methods in Fortran.* Englewood Cliffs, N. J. : Prentice-Hall, Inc.

Milton R. C. (1972). Computer evaluation of the multivariate normal integral. *Technometrics, 14(4):881-889.*

National Bureau of Standards. (1959). *Tables of the Bivariate Normal Distribution Function and Related Functions. Appl. Math. Ser. 50.* Washington, D. C.: U. S. Government Printing Office.

Pierre, D. A. and Lowe, M. J. (1975). *Mathematical Programming via Augmented Lagrangians: An Introduction With Computer Programs.* Reading, Mass.: Addison-Wesley.

Prekopa, A. (1970). On Probabilistic constrained programming. *Proc. Princeton Symp. Math. Programming, (H. W. Kuhn, ed.)* Princeton, N. J. : Princeton Univ. Press.

Ruben, H. (1960). Probability content of regions under spherical normal distributions, I. *Ann. Math. Statist., 31:598-618.*

Schlafli, L. (1858). On the multiple integral $\int dxdy\,dz$ whose limits are $p_1=a_1x+b_1y+\cdots+h_1z>0$, $p_2>0,\ldots,p_n>0$ and $x^2+y^2+\cdots+z^2<1$.

Sobel, M. and Uppuluri, V. R. R. (1972). On Bonferroni-type inequalities of the same degree for the probability of unions and intersections. *Ann. Math. Statist., 43:1549-1558.*

Steck, G. P. and Owen D. B. (1962). A note on the equicorrelated normal distribution. *Biometrika, 49:269-271.*

Takács, L. (1967). On the method of inclusion and exclusion. *J. Amer. Statist. Assoc., 62:102-113.*

APPROACHES FOR EMPIRICAL BAYES CONFIDENCE INTERVALS FOR A VECTOR OF EXPONENTIAL SCALE PARAMETERS

Bradley P. Carlin and Alan E. Gelfand, University of Connecticut

ABSTRACT

Parametric empirical Bayes methods of point estimation date to the landmark paper of James and Stein (1961). Interval estimation through parametric empirical Bayes techniques has a somewhat shorter history, which is summarized in the recent paper of Laird and Louis (1987). In the i.i.d. exchangeable case, one obtains a "naive" EB confidence interval by simply taking appropriate percentiles of the estimated posterior distribution of the parameter, where the estimation of the prior parameters ("hyperparameters") is accomplished through the marginal distribution of the data. Unfortunately, these "naive" intervals tend to be too short, since they fail to account for the variability in the estimation of the hyperparameters. That is, they don't attain the desired coverage probability, both in the classical sense and in the "EB" sense defined in Morris (1983a).

In this paper we consider two methods for developing EB intervals for exponential scale parameters which attempt to correct this deficiency in the naive intervals. The first is a "bias corrected naive" method inspired by Efron (1987). Simply put, this method adjusts the naive intervals using tail areas determined by the parametric structure of the model and the data. The second method uses a parametric bootstrap (Laird and Louis, 1987) to match a specified hyperprior Bayes solution. Finally, through simulation we compare methods with respect to EB coverage and length.

1. INTRODUCTION

Consider the i.i.d. exchangeable Bayesian formulation where $Y_1,\ldots,Y_p \sim \text{Gamma}(v_i,\beta_i)$, $i = 1,\ldots,p$ independent, v_i known and the β_i's have the conjugate inverse gamma (IG) prior, $\beta_1,\ldots,\beta_p \overset{iid}{\sim} IG(a,b)$, $a,b > 0$. We take $v_i = 1$ for convenience, though the case of general known v_i is discussed briefly in Section 2. Thus $f(y_i|\beta_i) = \beta_i^{-1}\exp(-y_i/\beta_i)$, $y_i > 0$, $\beta_i > 0$,

$g(\beta_i|a,b) = \exp(-1/\beta_i b)/(\Gamma(a)b^a\beta_i^{a+1})$, $a,b > 0$,

$i = 1,\ldots,p$. Hence the Y_i's are marginally i.i.d. with distribution

$$f(y_i|a,b) = ab/(by_i+1)^{a+1},\ y_i > 0 \qquad (1.1)$$

and the posterior distribution of β_i is $IG(a+1,(y_i+1/b)^{-1})$, i.e.,

$$f(\beta_i|y_i,a,b) = \frac{\exp(-y_i+1/b)/\beta_i)}{\Gamma(a+1)(y_i+1/b)^{-(a+1)}\beta_i^{a+2}}. \qquad (1.2)$$

Taking the scale parameter $b = 1$, we view a as unknown, and estimate it from the marginal distribution of $\underline{Y}$, $f(\underline{y}|a) = \prod_{i=1}^{p} f(y_i|a)$. For EB point estimation a best choice of a (e.g., MLE, UMVUE, moments estimator) is not clear. Not surprisingly, this same difficulty arises in developing EB confidence intervals. Usual estimators of a take the form $\hat{a}_c = c/\sum_{i=1}^{p}\log(Y_i+1)$. For instance, $c = p$ yields the MLE while $c = p-1$ yields the UMVUE. Choosing one of these as our estimate of a, the "naive" EB confidence interval for β_i is simply the upper and lower $\alpha/2$-points of the "estimated posterior," i.e., (1.2) with a replaced by $\hat{a}$. These intervals are called "naive" because in ignoring randomness in $\hat{a}$ they tend to be too short. More precisely, Morris (1983a,b) defines an EB confidence set of size $1-\alpha$ as a subset $t_\alpha(\underline{Y})$ of Θ such that

$$P(\theta \in t_\alpha(\underline{Y})) \geq 1 - \alpha \qquad (1.3)$$

where the probability is calculated over the joint distribution of θ and $\underline{Y}$. The naive intervals generally fail to satisfy (1.3).

In this paper, we propose two methods to correct this deficiency in the naive intervals. In Section 2 we introduce a method for bias-correcting the naive interval, and discuss some of its properties. Section 3 develops a method which matches any hyperprior Bayes solution. A parametric bootstrap (Laird and Louis, 1987) is used in place of numerical integration. Section 4 obtains simulated coverage probabilities and interval lengths for the methods.

2. THE BIAS CORRECTED NAIVE APPROACH

Efron (1987) proposed a general framework for correcting the bias in naive EB intervals. In our setting, a simpler bias correction may be developed. From (1.2) we see that the α^{th} quantile of the posterior distribution $f(\beta_i|y_i,a)$ is

$$q_\alpha(a) = 2(y_i+1)/D_{2(a+1)}^{-1}(1-\alpha) \qquad (2.1)$$

where D_k^{-1} is the inverse c.d.f. of a χ^2 distribution with k (not necessarily integer) degrees of freedom. Define

$$\begin{aligned}\alpha(\hat{a},a,\alpha) &= P\{\beta_i \leq q_\alpha(\hat{a})\,|\,\beta_i \sim f(\beta_i|y_i,a)\} \\ &= P\{\beta_i \leq 2(y_i+1)/D_{2(\hat{a}+1)}^{-1}(1-\alpha) \\ &\qquad |\,\beta_i \sim IG(a+1,(y_i+1)^{-1})\} \\ &= P\{\gamma_i \leq 2/D_{2(\hat{a}+1)}^{-1}(1-\alpha)\,|\,\gamma_i \sim IG(a+1,1)\}\end{aligned} \qquad (2.2)$$

where $\gamma_i = \beta_i/(y_i+1)$. Thus $\alpha(\hat{a},a,\alpha) = 1 - D_{2(a+1)}(D_{2(\hat{a}+1)}^{-1}(1-\alpha))$. Finally, let

$$\pi(a,\alpha) = E_{\hat{a}|a}[\rho(\hat{a},a,\alpha)]. \qquad (2.3)$$

For the UMVUE $\hat{a}$, $\hat{a}|a \sim IG(p,(a(p-1))^{-1})$.

Note that (2.3) is a naive EB tail area in the Morris sense, (1.3). Typically, $\pi(a,1-\alpha/2) - \pi(a,\alpha/2) < 1-\alpha$, that is, the naive intervals are too short. It is usually argued that this undercoverage arises because we are failing to take into account the variability in $\hat{a}$. In any event, suppose we solve

$$\pi(a,\alpha') = \alpha \qquad (2.4)$$

for α'. This α' would "correct the bias" in using $\hat{a}$ in our naive procedure. Applying (2.4) would produce intervals with exactly the desired coverage probability. But, of course, we can't solve (2.4) since a is unknown. Instead, we propose to solve

$$\pi(\hat{a},\alpha') = \alpha \qquad (2.5)$$

to obtain $\alpha' = \alpha'(\hat{a},\alpha)$. Then we take as our bias corrected naive EB confidence interval the naive interval with "α" replaced by "α'". Under mild regularity conditions, this procedure gives a unique confidence interval. We note that correcting α to $\alpha'(\hat{a},\alpha)$ is equivalent to correcting $q_\alpha(\hat{a})$ to $q_{\alpha'}(\hat{a})$, which in turn is equivalent to correcting the quantiles of the estimated posterior. Also note that since we were able to "scale out" y_i in (2.2), the integration in (1.3) could be done by integrating over $\hat{a}|a$. Equation (2.5) can be solved using a one-dimensional numerical integration (we transformed the IG to the interval (0,1) and used 16-point Gaussian integration--see Abramowitz and Stegun, 1967) with one rootfinder (using false position).

We can extend our work to the full Gamma/IG problem, i.e., $Y_i \overset{ind}{\sim} \text{Gamma}(v_i,\beta_i)$ where v_i known and $\beta_i \overset{iid}{\sim} IG(a,b)$, $i = 1,\ldots,p$. Again we take $b = 1$. (Note that this case includes the χ^2 scale problem.) One can show that $Y_i|a \sim \Gamma(v_i+a)/(\Gamma(v_i)\Gamma(a))\cdot y_i^{v_i-1}/(y_i+1)^{v_i+a}$, a Pearson Type VI distribution (Johnson and Kotz, 1970). Again we can scale out y_i as in (2.2), and now $\beta_i|a \sim IG(v_i+a,1)$. Note that now the MLE $\hat{a}$ is *no longer available in closed form.* However, since $T(\hat{a}) = \sum_{i=1}^{p} \log(y_i+1)$ is decreasing in $\hat{a}$, we can use the distribution of $T(\hat{a})$ to implement the bias correction.

Before concluding this section, we address the question of whether the bias correcting method actually produces EB confidence intervals in the Morris sense. Since from (2.5) α' is random, we need to look at the tail area

$$E_{\hat{a}|a}\{P(\beta_i \le q_{\alpha'(\hat{a},\alpha)}(\hat{a})|\beta_i \sim IG(a+1,1)\} \qquad (2.6)$$
$$= E_{\hat{a}|a}\rho(\hat{a},a,\alpha'(\hat{a},\alpha)).$$

While exact evaluation of this expectation is not possible, Carlin and Gelfand (1988) show that (2.6) falls in an interval containing α. In fact, (2.6) is bounded above by $\alpha + \max(I_1,I_2)$ and below by $\alpha + \min(I_1,I_2)$, where

$$I_1 = \int_{\hat{a}>a} [\alpha'(\hat{a},\alpha) - \rho(\hat{a},a,\alpha'(a,\alpha))]dF(\hat{a}|a),$$

and

$$I_2 = \int_{\hat{a}<a} [\alpha'(\hat{a},\alpha) - \rho(\hat{a},a,\alpha'(a,\alpha))]dF(\hat{a}|a).$$

Moreover, the simulations in Section 4 indicate that this method does achieve EB coverage, (1.3).

3. THE PARAMETRIC BOOTSTRAP APPROACH

Several authors (Deely and Lindley, 1981; Rubin, 1982; Morris, 1983a,b, 1987; Laird and Louis, 1987) in the PEB setting have attempted to account for the variation in estimating a hyperparameter by introducing a hyperprior distribution. Quantiles of the resulting "marginal posterior" are used in place of those of the estimated posterior. We note that while this approach is not directly aimed at developing intervals with the desired EB coverage, it is generally applicable and has worked well in our empirical studies. In our exponential/inverse gamma setting let us place a hyperprior $\tau(a)$ on a. This induces $h(a|\underline{y})$, which in turn induces

$$m(\underline{\beta}|\underline{y}) = \int \prod_{i=1}^{p} f(\beta_i|y_i,a)dH(a|\underline{y}), \qquad (3.1)$$

the marginal posterior. Using the MLE for a, which is sufficient for (1.1), and the flat hyperprior $\tau_1(a) = I_{(0,\infty)}(a)$, the marginal posterior for β_i simplifies to

$$m(\beta_i|y_i,\hat{a}) = \int f(\beta_i|y_i,a)dH(a|\hat{a})$$
$$= \int f(\beta_i|y_i,a)\cdot \text{Gamma}(p+1,\hat{a}/p)da. \qquad (3.2)$$

In this setting the Type III parametric bootstrap of Laird and Louis may be used to approximate (3.2). It calls for drawing β_i^* i.i.d. from $G(\beta|\hat{a})$, and then Y_i^* independently from $f(y|\beta_i^*)$, $i = 1,\ldots,p$. We then compute a^* from the Y_i^*'s in the same way that $\hat{a}$ was obtained from the Y_i's. Thus $a^*|\hat{a}$ is distributed as $F(a^*|\hat{a})$ $= IG(p,1/(\hat{a}p))$. Obtaining N bootstrapped a_j^*'s in this fashion, we use the mixture distribution

$$\sum_{j=1}^{N} f(\beta_i|y_i,a_j^*)/N = \sum_{j=1}^{N} IG(a_j^*+1,(y_i+1)^{-1})/N \qquad (3.3)$$

to approximate (3.2). The EB confidence interval for β_i is then computed by finding the $\alpha/2$ and $1-\alpha/2$ points of (3.3).

Note, however, that the expected value of (3.3) is

$$\int f(\beta_i|y_i,a^*)dF(a^*|\hat{a}). \qquad (3.4)$$

As Hill (1987) notes, $F(a^*|\hat{a})$ is not the same as $H(a|\hat{a})$, and thus (3.4) may be a poor approximation for (3.2). Again the empirical success of (3.3) suggests that this may not be an important issue, especially since the link between (3.2)

and desired EB coverage is tenuous. Nonetheless, how can we achieve a better approximation to (3.2) for our problem? Consider the integral

$$\int f(\beta_i|y_i, \hat{a}^2/a^*)\cdot(\hat{a}/a^*)\cdot f(a^*|\hat{a})da^* . \quad (3.5)$$

Next let $b^* = \hat{a}^2/a^*$, so that $b^*|\hat{a} \sim \text{Gamma}(p,\hat{a}/p)$. (Note that b^* is conditionally unbiased for $\hat{a}$.) Using this transformation, a little algebra shows that (3.5) is equal to

$$\int f(\beta_i|y_i,b^*)\cdot\text{Gamma}(p+1,\hat{a}/p)db^* \quad (3.6)$$

which is identical to (3.2). Thus, instead of (3.3), we would use

$$\sum_{j=1}^{N} IG(\hat{a}^2/a_j^*+1,(y_i+1)^{-1})\cdot(\hat{a}/a_j^*)/N \quad (3.7)$$

and take the upper and lower $\alpha/2$-points of this distribution as our confidence interval for β_i.

Ragunathan (1987) suggests a modification of (3.7) to a truly weighted average

$$\sum_{j=1}^{N} IG(\hat{a}^2/a_j^*+1,(y_i+1)^{-1})\cdot(\hat{a}/a_j^*)/\sum_{j=1}^{N}(\hat{a}/a_j^*). \quad (3.8)$$

Since $E(\hat{a}/a^*|\hat{a}) = 1$, this modification seems reasonable and works better computationally.

If instead of our flat hyperprior, we use $\tau_2(a) = 1/a \cdot I_{(0,\infty)}(a)$, then (3.2) becomes

$$\int f(\beta_i|y_i,a) \cdot \text{Gamma}(p,\hat{a}/p)da. \quad (3.9)$$

But since we know $b^*|\hat{a} \sim \text{Gamma}(p,\hat{a}/p)$, the $\hat{a}/a^*$ term in (3.5) is no longer needed. The approximation to (3.9) becomes

$$\sum_{j=1}^{N} IG(\hat{a}^2/a_j^* + 1, (y_i+1)^{-1})/N. \quad (3.10)$$

As we shall see, this approach proves better than (3.8). This is perhaps because τ_2 is a more appropriate hyperprior for a shape parameter; we are matching a more reasonable hyperprior Bayes solution.

As a final simplication, note that if we replace $\hat{a}^2/a_j^*$ by its expectation, (3.10) becomes

$$IG(\hat{a}+1,(y_i+1)^{-1}) \quad (3.11)$$

which is the estimated posterior. Thus not only are (3.7), (3.8), and (3.10) approximations to (3.2), but we may also view them as ways to incorporate bootstrap variation into the naive EB intervals in an effort to "lengthen" them.

Again, while exact evaluation of the coverage probabilities of these intervals is not possible, we shall see in the simulation results of the next section that the bootstrap intervals do generally achieve the desired EB coverage.

4. SIMULATED COVERAGE PROBABILITIES AND INTERVAL LENGTHS

In this section we present and discuss the results of a simulation study which compares the methods for our Exponential/IG problem. Since we are working in the EB framework, coverage was evaluated in this context. That is, for fixed a and p, we generated β_i's i.i.d. as IG(a,1), and then generated the Y_i's independently as Exponential(β_i), i = 1,...,p. Each simulation is based on 3000 replications; for the methods requiring a bootstrap, we used N = 400 bootstrap trials per replication.

Tables 4.1 - 4.4 show lower endpoint, upper endpoint, interval length (all averaged over both i and the replications) and individual and simultaneous EB coverage probability for the classical, naive EB, bias corrected naive EB, Laird and Louis bootstrap, and hyperprior matching bootstrap methods (3.8) and (3.10) (corresponding to hyperpriors τ_1 and τ_2, respectively, for p = 5, 10, true a = 2, 5, and nominal individual coverage probabilities γ = .90 and .95. Recall that in the bias corrected method, the choice of this estimator $\hat{a}$ affects three parts of the procedure: the computation of the π function (2.3) (we need the distribution of $\hat{a}|a$), the actual solution of (2.5), and in the estimated posterior distribution. (The last of these three is the only place $\hat{a}$ shows up in the naive procedure.) In our simulation, for the naive and bias corrected naive we show results obtained using the marginal UMVUE $\hat{a} = (p-1)/\sum \log(y_i+1)$. Results (not shown) obtained using the marginal MLE $\hat{a} = p/\sum \log(y_i+1)$ gave longer (i.e., too conservative) bias corrected intervals (extending further to the right), but shorter naive intervals. For the three bootstrap methods, we also used the UMVUE for $\hat{a}$, but this time because the Laird and Louis MLE intervals generally failed to attain the nominal coverage probability. The two hyperprior matched intervals were insensitive to the choice of $\hat{a}$. This was what we expected, since we were matching the same hyperprior Bayes solution, regardless of the choice of $\hat{a}$.

In terms of comparing the methods, several observations can be made. As expected, the classical intervals faithfully achieve the desired coverages, but are unacceptably long--in one case (p = 10, a = 5, γ = .95) more than ten times longer than the better EB intervals. As has been noted by previous authors (Morris, 1983a,b; Laird and Louis, 1987), the naive EB intervals perform surprisingly well, especially for small a and large p. Yet in no case do they achieve the desired coverage; for small p and large a they are especially poor. The bias corrected naive intervals, on the other hand, are slightly conservative, though not significantly so (the average coverage probability numbers have a standard error of about .5%). In addition, the bias corrected intervals have lengths that are quite competitive with those for the two bootstrap methods shown, especially when γ = .90. The bootstrap methods also produce intervals that generally hit the desired coverage. Notice that the intervals based on matching the flat hyperprior τ_1 generally fail to achieve the desired coverage probability. By using the hyperprior

TABLE 4.1: a = 2, p = 5

Interval Method	Average Lower Endpoint	Average Upper Endpoint	Average Interval Length	Average Indivdual Cov. Prob.	Simul-taneous Cov. Prob.
γ = .90					
Classical	.335	19.5	19.2	90.1	59.1
Naive EB	.355	3.87	3.51	83.9	46.8
Bias Corrected	.331	4.74	4.41	89.7	60.8
Laird and Louis	.339	5.15	4.81	90.4	63.2
τ_1 Matching	.287	3.23	2.95	86.8	55.6
τ_2 Matching	.311	4.00	3.69	89.4	61.0
γ = .95					
Classical	.268	39.1	38.8	95.2	78.3
Naive EB	.306	5.53	5.22	90.0	64.4
Bias Corrected	.285	7.84	7.55	95.2	79.0
Laird and Louis	.283	7.79	7.50	95.4	80.9
τ_1 Matching	.246	4.46	4.51	93.0	74.7
τ_2 Matching	.265	5.93	5.66	95.1	79.8

TABLE 4.2: a = 5, p = 5

Interval Method	Average Lower Endpoint	Average Upper Endpoint	Average Interval Length	Average Indivdual Cov. Prob.	Simul-taneous Cov. Prob.
γ = .90					
Classical	.084	4.89	4.81	89.9	59.0
Naive EB	.134	.690	.556	77.1	35.4
Bias Corrected	.116	1.03	.914	90.2	66.1
Laird and Louis	.114	1.04	.928	89.9	66.9
τ_1 Matching	.092	.620	.528	86.3	61.1
τ_2 Matching	.102	.810	.708	90.1	67.9
γ = .95					
Classical	.068	9.87	9.81	94.8	76.1
Naive EB	.120	.859	.739	84.6	52.4
Bias Corrected	.103	1.67	1.57	95.6	82.8
Laird and Louis	.096	1.41	1.31	95.1	81.3
τ_1 Matching	.081	.816	.735	91.8	75.7
τ_2 Matching	.089	1.10	1.01	94.7	81.1

TABLE 4.3: a = 2, p = 10

Interval Method	Average Lower Endpoint	Average Upper Endpoint	Average Interval Length	Average Indivdual Cov. Prob.	Simul-taneous Cov. Prob.
γ = .90					
Classical	.324	18.9	18.5	90.1	35.1
Naive EB	.341	3.20	2.86	87.3	27.9
Bias Corrected	.327	3.56	3.23	90.2	37.3
Laird and Louis	.320	3.53	3.21	90.4	37.9
τ_1 Matching	.294	2.77	2.48	88.6	33.3
τ_2 Matching	.307	3.11	2.80	89.8	36.4
γ = .95					
Classical	.262	38.2	37.9	95.0	59.3
Naive EB	.295	4.42	4.12	93.0	50.6
Bias Corrected	.283	5.23	4.94	95.3	62.4
Laird and Louis	.276	5.07	4.79	95.3	62.7
τ_1 Matching	.260	3.98	3.72	94.2	58.8
τ_2 Matching	.265	4.42	4.16	94.9	60.8

TABLE 4.4: a = 5, p = 10

Interval Method	Average Lower Endpoint	Average Upper Endpoint	Average Interval Length	Average Indivdual Cov. Prob.	Simultaneous Cov. Prob.
γ = .90					
Classical	.083	4.86	4.78	90.0	35.5
Naive EB	.127	.577	.450	82.9	21.2
Bias Corrected	.116	.710	.594	90.3	41.7
Laird and Louis	.115	.714	.599	90.6	44.2
τ_1 Matching	.103	.555	.452	88.5	40.0
τ_2 Matching	.109	.633	.524	90.1	43.0
γ = .95					
Classical	.068	9.82	9.76	94.9	59.1
Naive EB	.114	.701	.587	89.5	40.8
Bias Corrected	.104	.956	.853	95.3	67.0
Laird and Louis	.101	.914	.814	95.1	66.3
τ_1 Matching	.091	.701	.610	93.5	62.6
τ_2 Matching	.097	.811	.714	94.8	66.2

τ_2 which puts more weight on small values of a, these intervals are shifted to the right and now attain the nominal EB coverage level. Finally, note that these last intervals are also remarkably short. For example, when p = 5, a = 2, and γ = .95, the τ_2 matching intervals attain the desired coverage on the average, yet are only 74% as long on the average as the Laird and Louis intervals.

5. CONCLUSION

In this paper we have developed two methods for computing empirical Bayes confidence intervals for a vector of exponential scale parameters that take into account the uncertainty in estimating hyperparameter a. We have defined and illustrated a method of bias correcting the usual naive EB intervals, and also given a bootstrap method by which we can match a hyperprior Bayes solution, with associated approximations. Our simulation study indicates that the bias corrected naive method is a strong candidate, and also that modifying the Laird and Louis Type III bootstrap to approximate a different marginal posterior can offer substantial improvement in interval length without sacrificing coverage probability.

REFERENCES

1. Abramowitz, M., and Stegun, I. (1967), Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables (Applied Mathematics Series, 55), Washington, D.C.: National Bureau of Standards.

2. Carlin, B.P., and Gelfand, A.E. (1988), "Approaches for Empirical Bayes Confidence Intervals With Applications," Technical Report 88-1, University of Connecticut, Department of Statistics.

3. Deely, J.J., and Lindley, D.V. (1981), "Bayes Empirical Bayes," Journal of the American Statistical Association, 76, 833-841.

4. Efron, B. (1987), Comment on "Empirical Bayes Confidence Intervals Based on Bootstrap Samples," by N.M. Laird and T.A. Louis, Journal of the American Statistical Association, 82, 754.

5. Hill, J.R. (1987), Comment on "Empirical Bayes Confidence Intervals Based on Bootstrap Samples," by N.M. Laird and T.A. Louis, Journal of the American Statistifal Association, 82, 752-754.

6. James, W., and Stein, C. (1961), "Estimation With Quadratic Loss," in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1), Berkeley: University of California Press, 361-379.

7. Johnson, N.L., and Kotz, S. (1970), Distributions in Statistics: Continuous Univariate Distributions - 2, New York: John Wiley & Sons.

8. Laird, N.M., and Louis, T.A. (1987), "Empirical Bayes Confidence Intervals Based on Bootstrap Samples," Journal of the American Statistical Association, 82, 739-750.

9. Morris, C.N. (1983a), "Parametric Empirical Bayes Inference: Theory and Applications," Journal of the American Statistical Association, 78, 47-59.

10. _____ (1983b), "Parametric Empirical Bayes Confidence Intervals," in Scientific Inference, Data Analysis, and Robustness, New York: Academic Press, 25-50.

11. _____ (1987), "Determining the Accuracy of Bayesian Empirical Bayes Estimates in the Familiar Exponential Families," Technical Report 46, University of Texas, Center for Statistical Sciences.

12. Ragunathan, T.E. (1987), "Bootstrap Bayes," Technical Report, University of Washington, Department of Biostatistics.

A DATA ANALYSIS AND BAYESIAN FRAMEWORK FOR ERRORS-IN-VARIABLES

John H. Herbert, Energy Information Administration
U.S. Department of Energy, Washington, D.C. 20585

1. INTRODUCTION

If observed variables X_1, X_2, and X_3 are thought to be best represented in a regression analysis as a sum of a true variable value (X^*) and an unobserved random measurement error value (X^{**});

$$X = X^* + X^{**}, \tag{1}$$

then a relatively simple procedure is available for estimating the appropriate regression coefficients. The first step involves merely estimating three regression equations:

$$X_1 = a_1 + b_{12}X_2 + b_{13}X_3 + \ldots + b_{1n}X_n \tag{2}$$

$$X_2 = a_2 + b_{21}X_1 + b_{23}X_3 + \ldots + b_{2n}X_n \tag{3}$$

$$X_3 = a_3 + b_{31}X_1 + b_{32}X_2 + \ldots + b_{3n}X_n. \tag{4}$$

The a_i and b_{ij} are standard OLS coefficients.

Equations are next reexpressed in terms of one of the relationships of interest. To obtain the required set of coefficients for the relationship between variable 1 and variables 2 and 3, we move X_1 to the left hand side of the equals sign in (3) and (4) and solve for X_1; that is,

$$X_1 = + b_{12}X_2 + b_{13}X_3 \ldots$$

$$= + B^1{}_{12}X_2 + B^1{}_{13}X_3 \ldots \tag{2a}$$

$$X_1 = + (1/-b_{21})X_2 + (b_{23}/-b_{21})X_3 \ldots$$

$$= - B^2{}_{12}X_2 - B^2{}_{13}X_3 \ldots \tag{3a}$$

$$X_1 = + (b_{32}/-b_{31})X_2 + (-1/-b_{31})X_3 \ldots$$

$$= - B^3{}_{12}X_2 + B^3{}_{13}X_3 \ldots . \tag{4a}$$

Klepper and Leamer (1984) have shown that sets of coefficients such as $B^k{}_{12}$, where k(k = 1,2,3) is a direction of minimization, are maximum likelihood bounds for the relationship between variables 1 and 2, under the assumption that true variables and unobserved random error variables are normally distributed. They have demonstrated this for multiple regressions with several independent variables containing a random error.

In this paper we will first discuss the intellectual heritage of the procedure. Next, a simple method of estimating the regression coefficient bounds is set forth. Collinearity diagnostics, which are obtained directly as part of this procedure, are noted. Frisch's regression strategy is then discussed and applied to a previously published regression analysis. Finally a strategy for an errors-in-variables regression analysis, which is inherently Bayesian, is used to reduce the bounds for the estimated coefficients.

2. BACKGROUND

The recommended approach was discussed and applied at length by Ragnar Frisch (1934). Although the first editor of Econometrica, Frisch was more of a data analyst than an econometrician, as these terms are understood today. He generally believed that the the assumption of error-free independent variables required in Fisher's maximum likelihood approach to least-squares was highly unlikely to be encountered in applications with economic data. Fisher's approach, however, was being increasingly accepted by avant-garde econometricians such as Koopmans (1937) because it supplied an elegant formal framework for a regression analysis, and because it lent itself well to forecasting.

Haavelmo (1944, pp. 52-55) suggested that the Frisch approach to regression analysis was, in fact, appropriate as a general stochastic representation for certain types of economic behavior, not just as a method of evaluating the consequences of errors-in-variables in a standard regression analysis. Malinvaud (1981) noted that the estimation of regression coefficient bounds recommended by Frisch received little attention in the past because it imposed a computational burden and, more importantly, because it was originally developed in a non-stochastic setting as a data analysis tool. With modern computers and software it is no longer computationally burdensome. Kendall and Stuart (1979, pp. 379-380), Patefield (1981), Kalman (1982), Klepper and Leamer (1984), and Becker et al. (1985) have variously derived the bounds within well-defined stochastic contexts for a variety of regression models. Since there are many examples of regression analyses in the social and physical sciences in which many of the variables in a regression analysis are best modelled as containing a random error, this change in circumstances recommends increased application of this relatively simple procedure.

3. THE COMPUTATIONAL PROCEDURE

To obtain the required coefficient bounds, the collinearity indices, and the coordinate values for displaying the bounds graphically, a matrix of cofactors R' is computed for a correlation matrix R of all the variables of interest in the analysis;

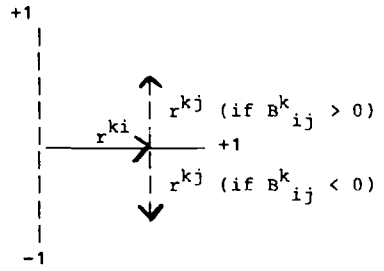
$$R = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & & \vdots \\ r_{n1} & \cdots & r_{nn} \end{bmatrix} \qquad R' = \begin{bmatrix} r^{11} & \cdots & r^{1n} \\ \vdots & & \vdots \\ r^{n1} & \cdots & r^{nn} \end{bmatrix}$$

When the elements of R' matrices are presented in tabular form Frisch designated these matrices as tilling tables. The regression coefficients are

calculated as ratios of elements in a tilling table with the following formula:

$$B^k_{ij} = -[r^{kj}/r^{ki}] \qquad (5)$$

The procedure used to construct the diagnostic graph is to move a a distance equal to the denominator value along the horizontal axis and then to move upwards or downwards a distance equal to the numerator value in (5), that is:



The number of bounds (k) is determined by the number of variables defined by 1. These graphs Frisch designated bunch maps. The extreme beams in a bunch map are regression coefficient bounds.

The diagonal elements in R' are the collinearity indices. They are related to the collinearity indices recently recommended by Stewart (1987) as ideal. Stewart interprets these indices within the context of the explanatory variables in a regression equation. Since in Frisch's scheme we are interested in the linear connection between all variables, not just the natural dependent variable, the collinearity indices are interpreted within the context of all the variables. The collinearity indices are readily shown to be related to the familiar multiple correlation coefficient (MCC) between variable i and the j, ... n other variables considered in the regression analysis, that is:

$$MCC_{i.j \ldots n} = \sqrt{1 - |R|/r^{ii}} \qquad (6)$$

where,

$|R|$ = the determinant of the matrix R

The inverse of $|R|/r^{ii}$ in (6) is the collinearity indice favored by Stewart (1987). Since $|R|$ is constant for any set of variables being evaluated, the comparison of several $r^{ii}/|R|$ is equivalent to the examinations of several r^{ii}. As an indication of collinearity problems, Frisch recommended examining the r^{ii}. For a particular correlation matrix, the greater the number of the r^{ii} that are similar in value, and the smaller the magnitude of the r^{ii}, the greater the chance that collinearity is a serious problem in obtaining reliable coefficients. Frisch also recommmended the examination of the ratios $r^{ii}/|R|$, not as collinearity indices, but as indicators of fit. The greater the difference in magnitude of an r^{ii} and an associated $|R|$, the greater the increase in the fit of a relationship by adding a variate.

4. THE PROBLEM

In two related articles by Herbert (1986, 1987a), a regression equation was estimated by OLS using state level annual data. The equation was evaluated by a battery of regression diagnostics as well as by a test of the hypothesis that state and temporal variance components were equal to 0. This latter test of the null hypothesis was not rejected.

The estimated regression equation expressed natural gas demand per customer in a period t (GD) as a linear combination of the price of gas (PG), the price of electricity (PE), income (Y), GD in the previous time period (GD(t-1)), and an indicator of average space heating requirements per customer (WH). The indicator WH was constructed based on changes between years in heating degree days and in the proportion of space heating customers among the customers in a state. All economic variables were expressed in constant dollars and all variables were expressed in logarithmic form. Except for WH, the estimated equation was a conventional econometric formulation of GD using state level/annual data (e.g. Beierlein et al. (1981) and Blattenberger et al. (1983)).

Additional analysis of the economic model by Herbert (1987b) indicated that WH represented the space heating capital stock portion of the capital stock surrogate variable GD(t-1) and this fact, along with the remaining variables included in the specification suggested dropping GD(t-1) from the regression equation and estimating;

$$GD = a - b_2PG + b_3PE + b_4Y + b_5WH + e \qquad (7)$$

where,

e = any random error associated with the behavioral relationship between GD and the other variables.

The expected sign for a coefficient is indicated by the designated sign for the coefficient in (7).

Additional examination of descriptive statistics of the Northeast region by Herbert (1988) and other regression analysis by Herbert (1986, 1987a), suggested that the major factors affecting GD during the time period had been considered. However, it was thought prudent to reconsider the use of a price of oil variable in (7) because fuel oil was widely used by households in the Northeast during the time period and other studies had recommended the use of this variable. However, estimated results from other studies were mixed with statistically insignificant coefficients frequently being estimated for the PO variable.

Additional analyses of the data by Doman et al. (1986), and Herbert (1987b) indicated that all variables included in the regression equation could probably be represented in the form of eq. 1. All variables are either proxy variables or they are known to be measured with error. Because of this measurement error problem, it was decided to evaluate the relationship between GD and PG, PE, Y, WH, and PO using the Frisch regression strategy.

5. THE EVALUATION OF NATURAL GAS DEMAND RELATIONSHIPS

The first series of bunch maps examined in Exhibit 1 indicate how the relationship between GD (labelled with a 1) and PG (labelled with a 2) is affected by the addition of the other variables in the analysis. Exhibit 2 lists the tilling tables required to construct the bunch maps presented in Exhibit 1.

Several of the tilling tables in Exhibit 2 will be used to construct the bunch maps reported in Exhibits 3 through 6. Additional tilling tables are listed in the Data Appendix. The order of a bunch map is read from left to right and from top to bottom.

The first bunch map in Exhibit 1 indicate that the relationsip between GD and PG is negative, as expected, whether we minimize in the direction of GD or PG and include only these two variables in the analysis. The range of the coefficients, however, is wide as indicated by the distance between the beams. These coefficients are readily calculable from the entries in the tilling table as being equal to -0.5 -(.5421/1) and -1.5 -(1/.5421).

The next two bunch maps in Exhibit 1 indicate that when we include PE (variable 3) and then Y (variable 4) in the analysis, the relationship is no longer identified (i.e. both positive and negative coefficient values are obtained). This is likely to occur when the overall fit of the linear relationship is poor and/or when we have not included a a sufficient number of variables to identify the relationship.

The inclusion of WH (variable 5), however, identifies the relationship between GD and PG. The range of likely values is obtained from entries in the first two columns of the fourth tilling table of Exhibit 1 with a lower bound of

Exhibit 1. Bunch Maps for the Relationship between Gas Demand (1) and Price of Gas (2)

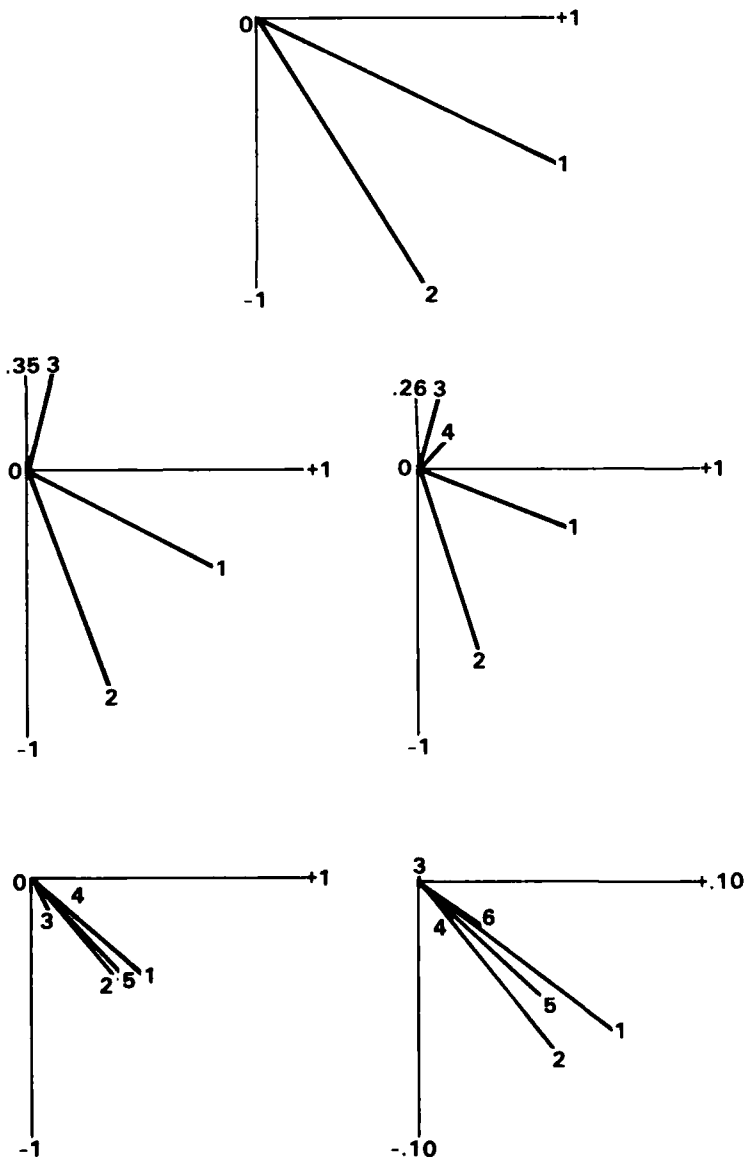


Exhibit 2. Tilling Tables Required to Construct the Bunch Maps in Exhibit 1

	1	2
1	1	.5421
2	.5421	1

	1	2	3
1	.6621	.2926	.1142
2	.2926	.8157	-.3436
3	.1142	-.3436	.7061

	1	2	3	4
1	.5302	.2036	.0803	.0856
2	.2036	.6678	-.2585	-.1250
3	.0803	-.2585	.5526	-.0442
4	.0856	-.1250	-.0442	.4345

	1	2	3	4	5
1	.3804	.2897	-.0529	-.0880	-.3218
2	.2897	.3333	-.1001	-.1096	-.2811
3	-.0529	-.1001	.1055	.0157	.0725
4	-.0880	-.1096	.0157	.1100	.1119
5	-.3218	-.2811	.0725	.1119	.3529

	1	2	3	4	5	6
1	.0665	.0533	.00003	-.0106	-.0460	-.0159
2	.0533	.0625	-.0046	-.0128	-.0371	.0216
3	.00003	-.0046	.0345	.0101	.0248	-.0316
4	-.0106	-.0128	.0101	.0213	.0240	-.0138
5	-.0460	-.0371	.0248	.0240	.0675	-.0238
6	-.0159	-.0216	-.0316	-.0138	-.0238	.0579

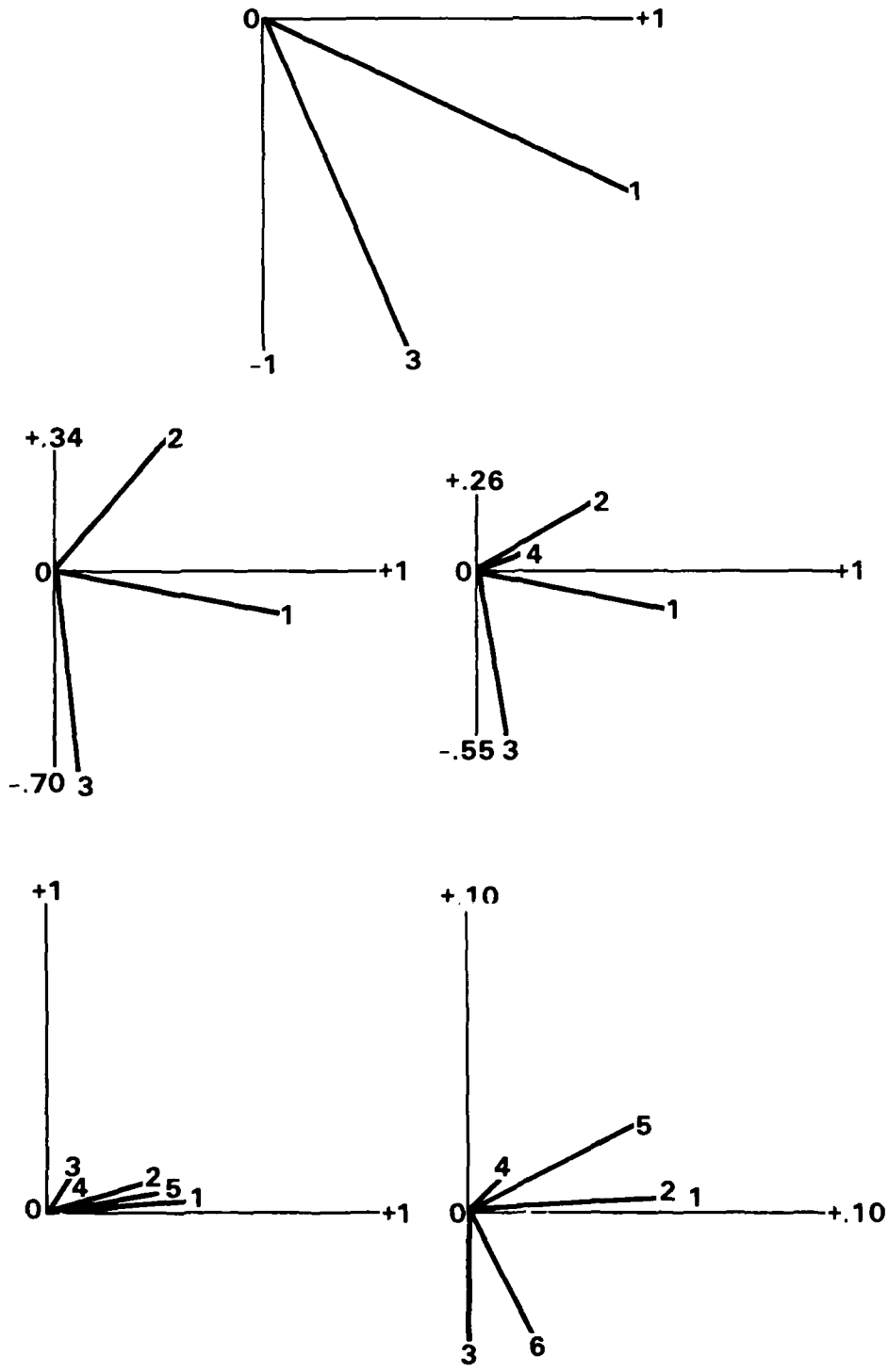


Exhibit 3. Bunch Map for the Relationship between Gas Demand (1) and the Price of Electricity (3).

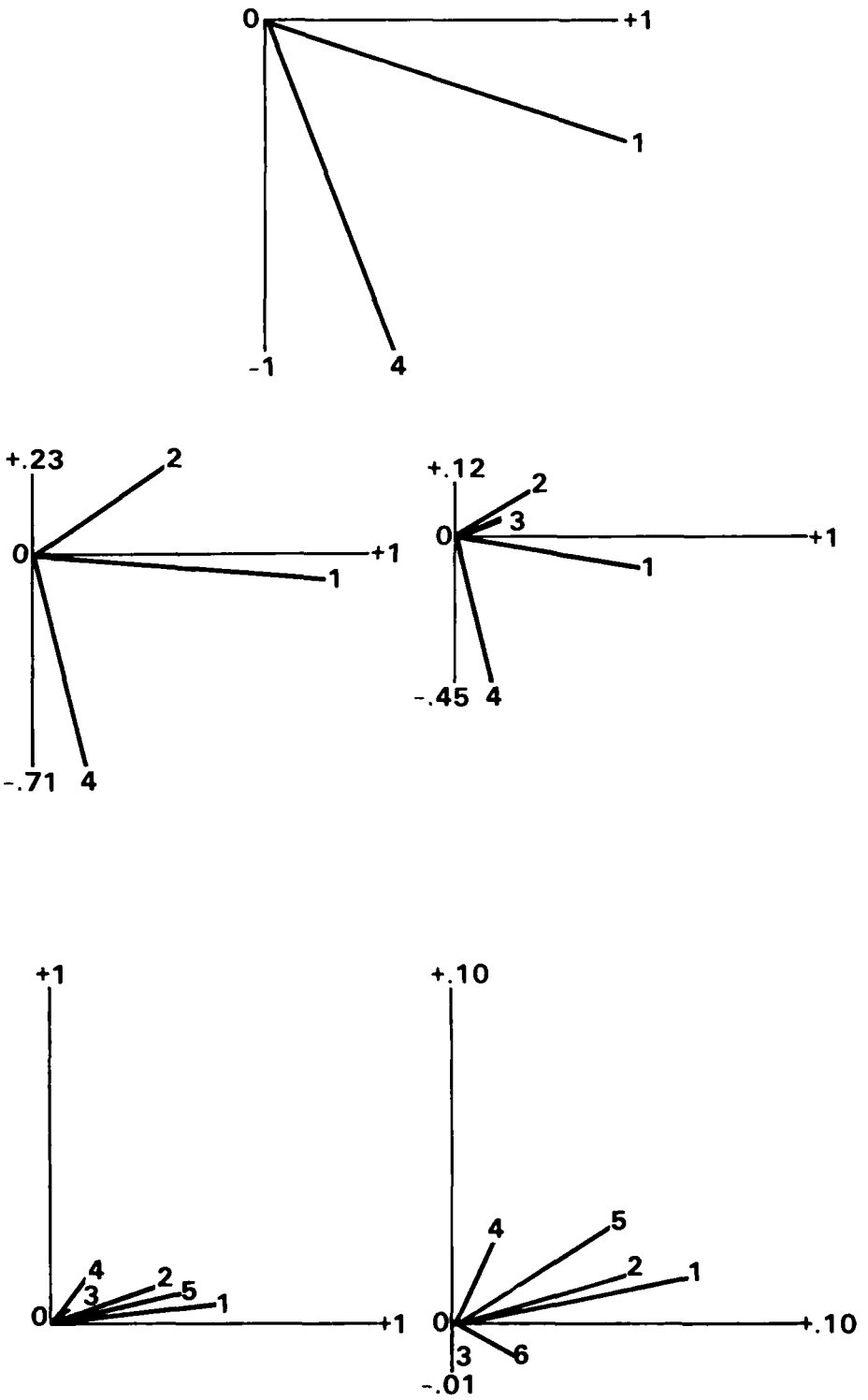


Exhibit 4. Bunch Map for the Relationship between Gas Demand (1) and Income (4).

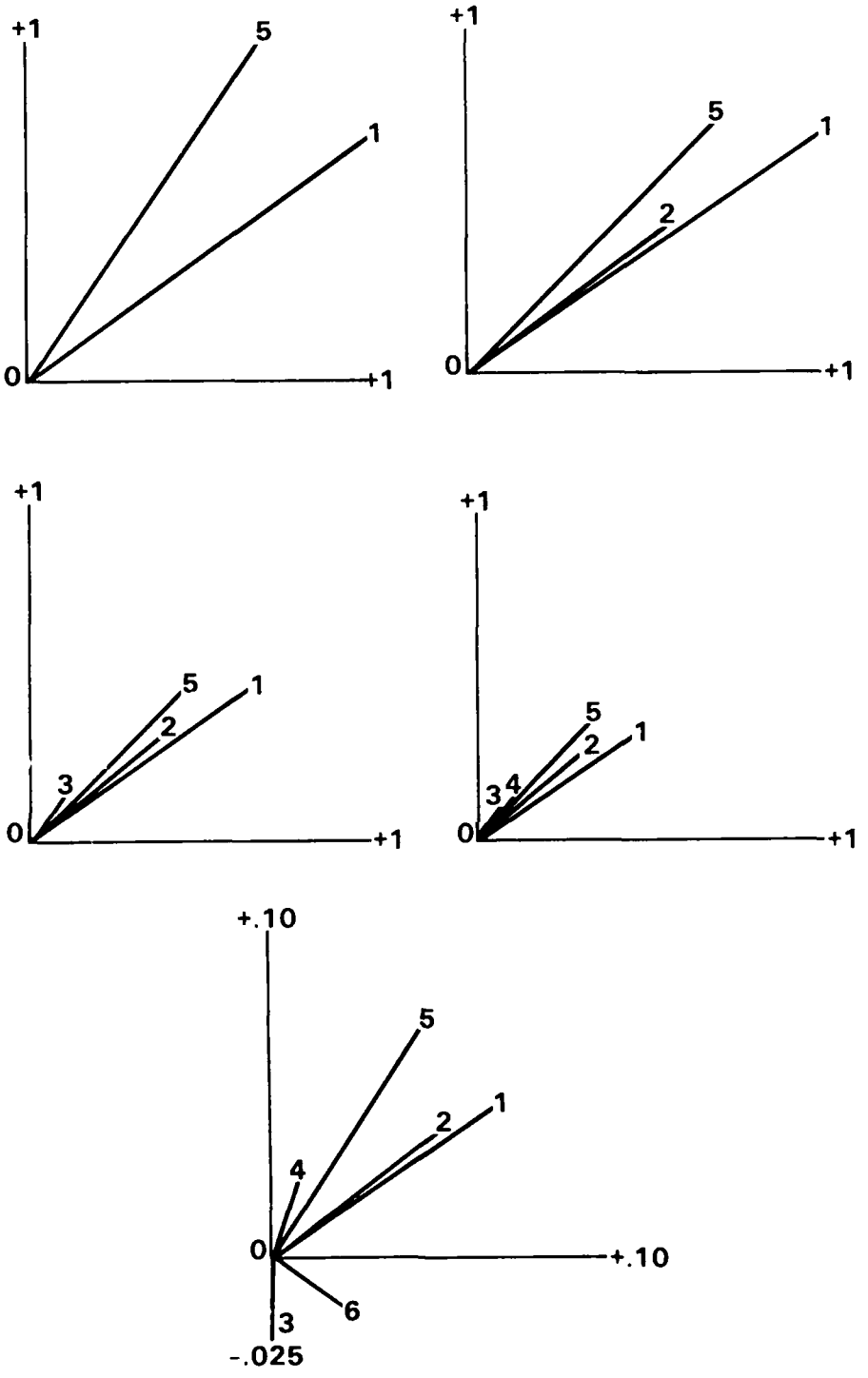


Exhibit 5. Bunch Map for the Relationship between Gas Demand (1) and Average Space Heating Requirements per Gas Customer (5).

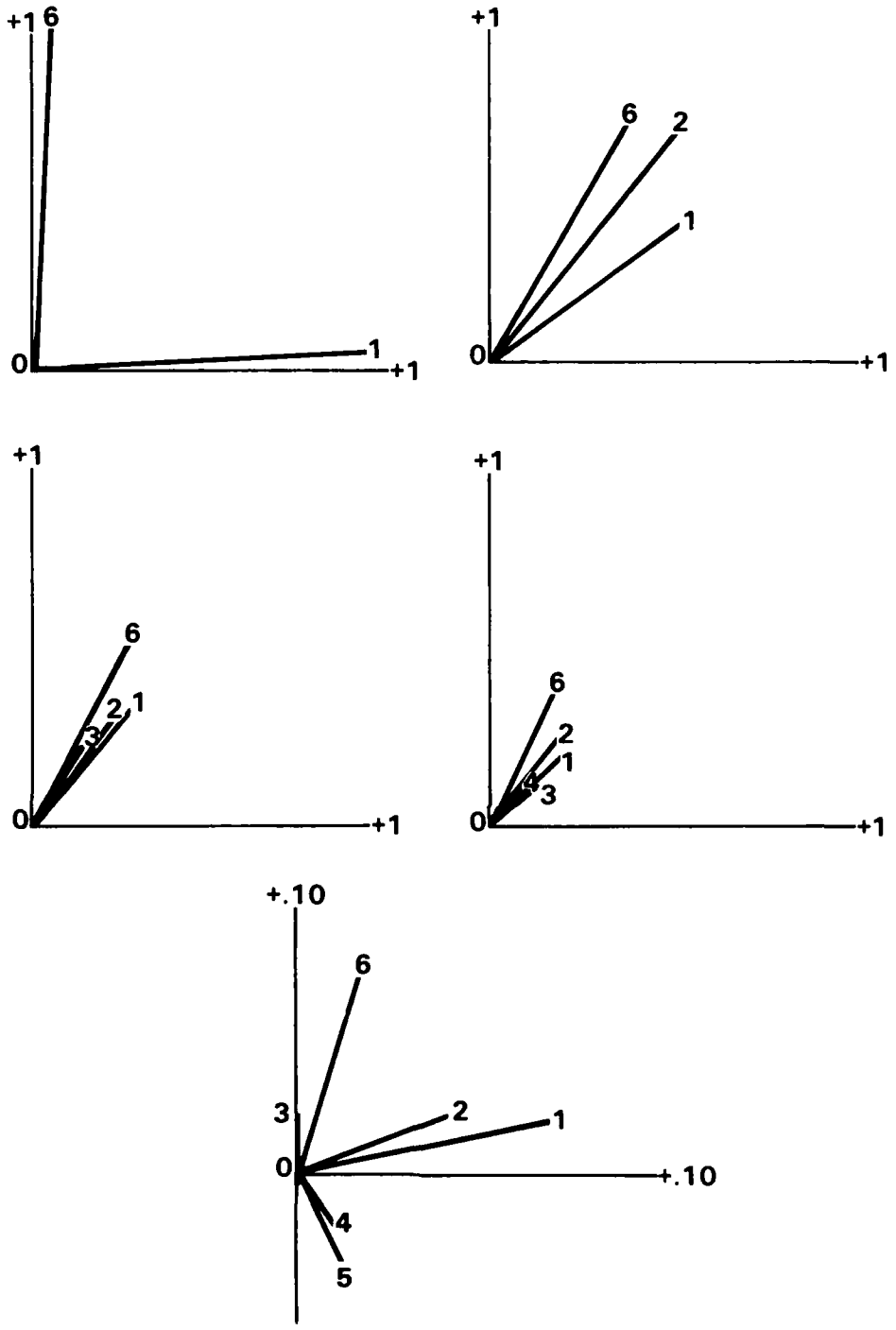


Exhibit 6. Bunch Map for the Relationship between Gas Demand (1) and the Price of Oil (6).

-.76 and an upper bound of -1.89. As a check on the linear connection between any one variable and the other variables for this bunch map, the multiple correlation coefficient is readily obtained from any entry along the diagonal of the associated tilling table and the last entry in the last tilling table which entry represents R" for the correlation matrix that includes variable 1 through 5. For example,

$$MCC_{GD.PG\ \ldots\ WH} = (1 - .0579/.3804)^{1/2} = .92$$

Clearly, the fit is improved by including variables 3 through 5 in the analysis. The increase in fit is also indicated by the length of the beams. The similarity of the diagonal elements for PE and Y in the associated tilling table indicates the similarity of the linear connection between these variables and the other variables.

The final bunch map in Exhibit 1 indicates that the relationship is no longer identified when we add PO (variable 6). The length of the axis had to be reduced from 1 to 0.10 for this bunch map to be properly viewed. The greatly reduced beam lengths when we add PO indicates that the variability for estimating any coefficient is greatly reduced when we include this variable in the analysis. Several of the entries in the tilling table used to construct these beams are so small that the coefficient is almost of the indeterminant form of a zero divided by a zero. According to Frisch (1934, pp 5-6), when the magnitudes used to estimate the coefficients are especially small, it is reasonable to consider such small magnitudes to be a consequence of randomness in the data from the unobservable random component. The coefficient B^3_{12} is a good example of such a case. The coefficient is calculated as the ratio of .0046/.00003 or 15.33, which is an unreasonably large value for this coefficient. The diagonal elements are also similar. This suggests that the linear connection between any one variable and the other variables is very similar and that collinearity is a problem because observed variables are measured with error. The calculation of the appropriate multiple correlation coefficients, for which the appropriate |R| is .005289, also reveals this similarity, they are:

$$MCC_{GD\ .\ PG\ .\ .\ .\ PO} = .9596$$
$$MCC_{PG\ .\ GD\ .\ .\ .\ PO} = .9570$$
$$MCC_{WH\ .\ GD\ .\ .\ .\ PO} = .9603$$
$$MCC_{PO\ .\ GD\ .\ .\ .\ WH} = .9504$$

The bunch maps for Exhibits 3 and 4 are similar in the sense that the relationship between GD and either Y or PE is identified when we include WH and not identified when we include the PO variable. Nonetheless, the range of likely values for the relationship for PE and Y is quite large for the bunch map that includes WH. For example, B^4_{13} is equal to .17, B^1_{13} is equal to 1.99.

Exhibit 5 displays a different picture from the preceding bunch map exhibits. The relationship is consistently identified and the range of likely values is relatively tight, until we add PO. In general, this relationship is well determined as long as PO is not included.

Exhibit 6 indicates that the relationship between gas demand and the price of oil is consistently positive, as expected for a cross price elasticity, until we add WH. the relationship is also fairly well-determined when we include only variables 1, 2, and 3. The relationship is less well determined when we add variable 4.

6. THE BAYESIAN TURN

In the preceding analysis, we have used the Frisch technique to: identify the range of likely values for a coefficient; to determine whether the identification of a relationship was affected by the addition of a particular variable in the analysis; to efficiently discover any possible collinearity problems in the data set; but we have not imposed knowledge of data in the estimation. Some information, however, is available on the relative magnitude of the measurement errors associated with each variable. For example, we expect the observed PO variable to be the least accurate measure of it's true variable value and we can impose such restrictions on the other variables. Moreover, information about measurement errors can be used to bound or reduce the range of likely values for a coefficient by the method proposed by Klepper and Leamer (1984) and by Klepper (1987). One can think of the procedure as indicating how the sample information would map different prior restrictions into posteriors. Rather than update a prior distribution for a regression coefficient based on sample information, the procedure indicates how the sample information would map different priors into posteriors. In order to implement the procedure, judgements must be formed either about the maximum value of R^2 associated with (7) when all variables are correctly measured and/or about measurement error variances of the explanatory variables. For this application the proportion of the observed variance in a measured variable due to measurement error is specified. The necessary assumptions and the nature of the measurement error in the observed variable values considered here are discussed in further detail in Herbert (1987c). Based on the previous analyses we specify the error variance (VAR) as a proportion of each observed variable variance to be:

$$VAR_{GD**}/VAR_{GD} = 0.02 \qquad (8)$$
$$VAR_{PG**}/VAR_{PG} = 0.02 \qquad (9)$$
$$VAR_{PE**}/VAR_{PE} = 0.03 \qquad (10)$$
$$VAR_{Y**}/VAR_{Y} = 0.03 \qquad (11)$$
$$VAR_{WH**}/VAR_{WH} = 0.04 \qquad (12)$$

The initial estimated relationships between GD and the other variables are obtained by first minimizing, as in (2) - (4), and then normalizing, as in (2a) - (4a). These estimates are reported in Exhibit 7. The final estimates, after we impose the measurement error variance constraints, as in (8) thru (12), are also reported. These final

EXHIBIT 7. Initial and Final Estimates

Coefficients	Minimization Direction				
Initial	GD	PG	PE	Y	WH
PG	-.76	-1.15	-1.89	-1.25	-1.15
PE	.14	.35	1.99	.80	.87
Y	.23	.38	.30	1.25	.35
WH	.83	.97	1.37	1.27	.91
Final					
PG	-.83	-1.04	-1.36	-1.08	-.90
PE	.18	.30	1.03	.21	.24
Y	.28	.36	.33	.79	.35
WH	.91	.99	1.17	1.14	1.06
Elasticities					
PG	-.56	-.76	-.92	-.73	-.61
PE	.21	.34	1.18	.24	.28
Y	.15	.19	.17	.41	.18
WH	.70	.76	.90	.89	.82

coefficients are also expressed as elasticities which are commonly used measures in economic analysis. The elasticities are obtained by multiplying the coefficients by the ratio of the appropriate standard errors, that is:

$$SB^k_{ij} = B^k_{ij} \cdot (S_i/S_j) = -(r^{kj}/S_j)/(r^{ki}/S_i) \quad (13)$$

This type of transformation is used to obtain the coefficient that would have been obtained if the sample variance/covariance matrix rather than the correlation matrix were used in the estimation of coefficients as is ordinarily done in most regression analyses.

A comparison of the entries in Exhibit 7 for the initial estimation and the final estimation indicates that the range of likely values in the final estimation is much reduced from the range of likely values in the original estimation. In particular the range of values for the PE coefficient is much narrower. Nonetheless, the upper bounds of 1.18 for the PE elasticity and of .41 for Y appear high relative to the other values for these coefficients. The elasticity for PE also seems high from a subject matter point of view.

The key coefficients in the analyses of gas demand are the PG coefficents, designated an own-price elasticity when it is expressed as an elasticity, and the WH coefficient which represents space heating capital equipment stock effects. Fortunately, the WH and PG coefficients are more stable across equations than the other coefficients.

7. SUMMARY AND CONCLUSIONS

In this paper we have demonstrated the instructive value of the Frisch regression strategy as a data analysis tool when errors-in-variables are suspected in a regression analysis. The Frisch strategy was used to discover any identification problems for the relationship between GD and the other variables. The WH variable was found to be important and the PO variable was found to be detrimental in the identification process. The instability of the Y and PE coefficients was clearly observable from a casual examination of the bunch maps. The analysis also suggests that information about the extent of the measurement error in Y and PE might be able to be used to reduce the range of likely values, that is to increase the precision, of the reported coefficents. For example, we would not have to minimize in the direction of PE if it were found that the PE variable accurately represented the price paid by natural gas customers for electricity. This would greatly reduce the reported bounds on the elasticity as indicated by reconstructing Exhibit 7 without the PE column. Frisch's recommended procedures were also used to identify any collinearity problems within the data set. This application does not circumscribe the usefulness of the Frisch strategy as a data analysis tool. For example, Malinvaud (1966, pp. 32-36) used the method as a means of identifying the number of linear relationship in a data set and then estimating these seperate relationships rather than one relationship. Frisch (1934) and Stone (1952), throughout their texts, used the Frisch strategy as a visual documentation of a regression analysis.

Finally, we have shown how newer techniques can help in the identification process. With these techniques, information about the measurement error in observed variables was used to reduce the range of likely values for estimated coefficients. This newer approach further underlines the importance of information on the measurement error for an estimation.

Acknowledgements:
The views expressed are those of the author and do not necessarily reflect those of the EIA, U. S. Department of Energy. The author would like to thank Steven Klepper of Carnegie Mellon University, Nancy Kirkendall of the EIA and Phil Kott of the National Agricultural Statistics Service, US Department of Agriculture, for comments and discussions.

References:

Beierlein J. G., J.W. Dunn and J. C. McConnon, (1981), "The Demand for Electricity and Natural Gas in the Northeastern United States," The Review of Economics and Statistics, 63, 403-408.

Bekker, P. A., T.J. Wansnbeek, and A. Kapteyn, (1985), "Errors in Variables in Econometrics: New Developments and Recurrent Themes," Statistica Neerlandica, 39, 129-141.

Blattenberger G. R., L. D. Taylor, and R. K. Rennhack, (1983), "Natural Gas Availability and the Residential Demand for Energy," The Energy Journal, 4, 23-45.

Doman, L. A., J. H. Herbert and R. Milller, (1986), An Assessment of the Quality of Selected EIA Data Series - Energy Consumption Data, Washington, D.C.: Energy Information Administration, U.S. Department of Energy.

Frisch, R. (1934b), Statistical Confluence Analysis by Means of Complete Regression Systems, Oslo, Norway: University Institute of Economics.

Haavelmo, T. (1944), The Probability Appraoch in Econometrics, Econometrica Supplement 12, iii-115.

Herbert, J. H. (1986), "Data Analysis, Specification, and Estimation of an Aggregate Relationship for Sales of Natural Gas per Customer," Journal of Economic and Social Measurement, 14, 165-174.

Herbert J. H. (1987a), "Data matters - Specification and

estimation of natural gas demand per customer in the Northeastern United States," Computational Statistics and Data Analysis, 5, 67-78.

Herbert, J. H. (1987b), "Measurement Error and the Estimation of Regression Equations - A Case Study," Proceedings of the Section on Economic and Business Statistics, American Statistical Association, 187-190.

Herbert, J. H. (1987c). "Demand for Natural Gas at the State Level Twenty Years of Effort," Review of Regional Studies, 17, 79-87.

Herbert, J. H. (1988), "Institutional and Economic Factors in Residential Gas Markets in the Northeastern United States between 1960-1984," Energy, 18, to be published.

Kalman, R. E. (1982), "System Identification from Noisy Data", in A. R. Bednarik and L. Cesari (eds.), Dynamical Systems II, New York: Academic Press.

Klepper, S. and E. E. Leamer (1984), "Consistent Sets of Estimates for Regressions with Errors in all Variables," Econometrica, 52, 163-183.

Klepper, S. (1987), Regressor Diagnostics for the Classical Errors-in-Variables Model, in Proceedings of the Section on Business and Economic Statistics, American Statistical Association, to be published.

Koopmans, T. (1937), Linear Regression Analysis of Economic Time Series, Haarlem, Netherlands.

Malinvaud, E. (1966), Statistical Methods of Econometrics, 1st Edition, New York: North Holland Publishing Company.

Malinvaud, E. (1981), Statistical Methods of Econometrics, 3rd Edition, New York: North Holland Publishing Company.

Patefield, W. M. (1981), "Multivariate Linear Relationships: Maximum Likelihood Estimation and Regression Bounds," Journal of the Royal Statistical Society B, 43, 342-352.

Stewart, G. W., (1987), "Collinearity and Least Squares Regression," Statistical Science, 2, 68-84.

Stone, R. (1954), The Measurement of Consumers' Expenditure and Behavior in the United Kingdom, 1920-1938, Cambridge: Cambridge University Press.

DATA APPENDIX

This appendix contains some additional statistics which are useful in considering the results presented in the main body of the text. Listed below for the convenience of some readers is the standard estimated results (standard errors are presented in parenthesis) for the initial equation estimated by OLS.

$$GD = -2.0 - .52PG + .16PE + .44Y + .65WH$$
$$\quad\quad (.5) \quad (.04) \quad (.07) \quad (.11) \quad (.04)$$

The tilling tables required to compute the coefficents in Exhibits 3 through 6, which were not presented in Table 2, are presented in Exhibit 8. Additional tilling tables relating to the controversial PO variable are reported in Exhibit 9. The standard deviations of the variables used in this analysis are: GD=.16097, PG=.23739, PE=.14041, Y=.08394, WH=.20928 which are required for the calculation of the elasticities. The average values of the variables used in this analysis are: GD=4.55940, PG=.43313, PE= 1.94563, Y=1.27398, WH=9.07288.

Exhibit 8. Tilling Tables in Addition to Those in Exhibit 2 Required to Construct Bunch Maps

	1	3
1	1	.429
3	.429	1

	1	4
1	1	.379
4	.379	1

	1	5
1	1	-.688
5	-.688	1

	1	6
1	1	-.031
6	-.031	1

	1	2	4
1	.812	.378	.144
2	.378	.856	-.228
4	.144	-.228	.706

	1	2	5
1	.998	.573	-.713
2	.573	.526	-.418
5	-.713	-.418	.706

	1	2	6
1	.560	.564	-.391
2	.563	.999	-.680
6	-.391	-.680	.706

	1	2	3	5
1	.5893	.3840	-.0767	-.4416
2	.3840	.4260	-.1605	-.3224
3	-.0767	-.1605	.1963	.1074
5	-.4416	-.3224	.1074	.4545

	1	2	3	6
1	.3129	.2677	.1579	-.2862
2	.2677	.4060	.0868	-.3320
3	.1579	.0868	.2426	-.2144
6	-.2862	-.3320	-.2145	.4545

	1	2	3	4	6
1	.2505	.1997	.1212	.0413	-.2295
2	.1997	.3009	.0646	.0030	-.2473
3	.1212	.0646	.1807	.0092	-.1629
4	.0413	.0030	.0092	.0910	-.0383
6	-.2295	-.2473	-.1628	-.0383	.3529

Exhibit 9. Tilling Tables for Examining the Relationship Between Price of Oil (6) and Other Variables

	2	3	4
2	.886	-.435	-.237
3	-.435	.812	-.086
4	-.237	-.086	.662

	2	3	6
2	.608	-.166	-.299
3	-.166	.560	-.240
6	-.299	-.240	.662

	2	4	6
2	.896	-.220	-.524
4	-.220	.860	-.034
6	-.524	-.034	.812

	2	3	4	6
2	.57	-.12	-.11	-.24
3	-.12	.45	-.04	-.19
4	-.11	-.04	.31	-.002
6	-.24	-.19	-.002	.53

	2	3	4	5
2	.74	-.39	-.28	-.24
3	-.39	.64	.02	.18
4	-.28	.02	.59	.25
5	-.24	.18	.25	.53

	1	3	4	6
1	.53	.35	.13	-.29
3	.35	.74	.04	-.49
4	.13	.04	.41	-.16
6	-.29	-.49	-.16	.67

	2	3	4	5	6
2	.140	-.032	-.030	-.001	-.062
3	-.032	.243	.071	.175	-.222
4	-.030	.071	.138	.117	-.115
5	-.001	.175	.117	.251	-.245
6	-.062	-.222	-.115	-.245	.380

	1	2	4	5	6
1	.243	.194	-.039	-.168	-.058
2	.194	.226	-.042	-.123	-.094
4	-.039	-.042	.067	.061	-.017
5	-.168	-.123	.061	.181	-.004
6	-.058	-.094	-.017	-.004	.105

	1	3	4	5	6
1	.190	.026	.002	-.096	.016
3	.026	.226	.061	.146	-.219
4	.002	.061	.123	.108	-.121
5	-.096	.146	.108	.301	.292
6	.016	-.219	-.121	.292	.333

	1	2	3	5	6
1	.138	.106	.011	-.077	-.055
2	.106	.123	.003	-.057	-.067
3	.011	.003	.067	.030	-.056
5	-.077	-.057	.030	.091	-.018
6	-.055	-.067	-.056	-.018	.109

THE EFFECT OF LOW COVARIATE_CRITERION CORRELATIONS ON THE ANALYSIS-OF-COVARIANCE

Michael J. Rovine, Alexander von Eye, and Phillip Wood
The Pennsylvania State University

Abstract

Analysis of covariance under conditions of small covariate-criterion correlations is examined. In the case of correlational research the increase in precision of an F-test assumed by the addition of covariates is questioned. To test whether the increase in precision assumed was offset by an increase in bias, a set of simulations was conducted. The first simulation showed the degree to which non-zero non-significant correlations between covariate and criterion changed the tail probability of the F-test. The second simulation included all "significant" covariates from a set of random normal variates showing how selecting all significant covariates without controlling for the number of covariates considered can effect the F-test.

Introduction

Researchers faced with results of an analysis of variance (ANOVA) often wonder how the analysis would have changed had study participants been equivalent on background variables. Attempts to control for effects of these background or nuisance variables in such uncontrolled studies has led to widespread use of analysis of covariance (ANCOVA). While most scientists agree this procedure is a poor substitute for true experimental control, many feel that certain research designs (Cohen & Cohen, 1975), in particular, non-equivalent group designs often used in the situations described above, require the use of a covariate or a set of covariates; especially in situations in which linear univariate relationships between background variables are explanations for any obtained group differences on a criterion variable of interest. In a controlled study the possible confound represented by a specific background variable is often randomized out of the design; however, in the uncontrolled study, some other procedure, most often ANCOVA is suggested to adjust for inequalities on that background variable.

While some conditions under which ANCOVA may be troublesome have been addressed (Elashoff, 1969; Glass, Peckham, & Sanders, 1972), one problem under-represented in the literature regards the use of a variable or a set of variables as covariates when those variables have relatively low correlations with the criterion variable under study. This paper shows the degree that such low correlations can bias the conclusions based on tests of effects of ANCOVA. This bias is due to assumptions underlying ANCOVA. Simulated and real data examples are presented. Finally, a Monte Carlo study showing the effects of selecting and controlling for significant covariates generated from a set of random variates will be presented. As in most real data analyses, the correlation between covariates and independent variables is not controlled.

On the Use of ANCOVA

In the analysis of variance, any differences on the criterion variable are assumed to be due to membership in the groups defined by the independent grouping or treatment variable. Let $E(Y_1)$ and $E(Y_2)$ be the expectations or means for Groups 1 and 2, respectively, then

$$E(Y_1) - E(Y_2) = \alpha. \quad (1)$$

If the difference is a function of only group or treatment effects, the estimate of α is unbiased. If, on the other hand, the difference in the observed values on the criterion may result from contributions of other sources in addition to the independent variable, Equation 1 becomes

$$E(Y_1) - E(Y_2) = \alpha + f(X_1, X_2, X_3, \ldots, X_k) \quad (2)$$

or expressed as a conditional probability

$$E(Y_1|X_1, \ldots, X_k) - E(Y_2|X_1, \ldots X_k) = \alpha \quad (3)$$

where f is a function of some set of variables X_i that contribute to the observed group differences. This function, then, represents the degree of bias in the analysis of variances when sources of systematic criterion variable differences other than the planned independent variable exist.

When a single variable X can be located that provides an unbiased estimator of α, Equation 2 can be written in terms of conditional probabilities as

$$E(Y_1|X) - E(Y_2|X) = \alpha \quad (4)$$

This covariate, X, then, allows one to adjust the analysis for this additional source of variation. Analysis-of-covariance was originally developed to increase precision in randomized experiments by adjusting for effects of additional variables not involved in the assignment of individuals to treatment or control groups (Fisher, 1932). Adjustments made using the covariate are only expected to yield unbiased estimates of effects (group differences) in the case of random assignment.

Compared with ANOVA, ANCOVA is assumed to provide a better, though possibly biased, estimate of α when a covariate can be isolated that is confounded with the treatment. With such a confound ANCOVA provides a more precise error term (Cochran, 1957). The benefit due to increased precision may be offset by bias introduced into the analyses by addition of a covariate. Although the degree of bias in uncontrolled studies is often unknown (Weisberg, 1979), analysis of covariance is suggested as an appropriate data analytic strategy when sources of variation are located that are related to the dependent or criterion variable of interest but are unrelated to the independent grouping variables representing treatment or individual difference effects.

The size of relationship between any covariate and the criterion necessary to minimize bias and maximize precision has normally not been specified, although some suggestions have appeared. Cox (1957) suggested that when both covariate and criterion are assumed to be drawn from a bivariate normal distribution, $\rho > .60$ could be used as a cutoff when ANCOVA is preferable to blocking. Maxwell, Delaney, and Dill (1984) argued that the size of the correlation is generally not important in deciding between blocking and use of a continuous covariate. In designs such as the

non-equivalent post-test design in which the covariate is used as a proxy pretest, only a perfect correlation between the covariate and criterion can assure that the use of a covariate as a control would not introduce bias (Cook & Campbell, 1979). These and similar suggestions (Myers, 1979) make one wonder what happens in the case when ρ tends toward zero. In this vein, the low correlation will be discussed in the context of assumptions of ANCOVA.

As often discussed (Cochran, 1957; Myers, 1979), ANCOVA is designed for the analysis of an experiment in which a set of nT individuals have been selected at random and assigned at random to T treatment conditions. The complication arises when another variable, X, is shown to be correlated with the criterion variable measuring the way in which manipulated groups are expected to differ. The question then arises: Did the groups really differ on the criterion because of the manipulation, or can that difference be explained by the covariate? The way usually recommended to answer the questions involves adjustment of scores of the dependent variable by extracting the effect of the covariate from the criterion and essentially analyzing residual scores. If a difference is still obtained, the treatment groups are more likely to actually differ on the criterion variable.

In order to properly implement the ANCOVA a set of assumptions is required to assure that tests of differences in mean criterion scores are unbiased (Elashoff, 1969).

The model for the simple one-way fixed effects analysis of covariance is

$$Y_i = \mu + \alpha_j + \beta (X_i - \bar{X}) + e_i \qquad (5)$$

in which μ is the mean of the criterion variable across individuals and treatments, α_j is the deviation from the mean due to the effect of the treatment, $\beta(X_i - \bar{X})$ is the variability accounted for by the covariate expressed in terms of the regression slope, β, of Y_i onto $(X_i - \bar{X})$, and e_i is an unexplained error in the individual's residual score. The ANCOVA model is an extension of the ANOVA model and is subject to the same assumptions with

$$e_i \sim NID(0, \sigma_e^2) \qquad \text{for } i = 1, \ldots, nT \qquad (6)$$

$$\Sigma \alpha_j = 0 \qquad \text{for } j = 1, \ldots, k \qquad (7)$$

when individuals are randomly assigned to the treatment conditions as required by the ANOVA.

Additional assumptions that are needed when a covariate is included in the analysis appear in Elashoff (1969). Violation of any ANCOVA assumptions can introduce bias into tests of effects. In particular, use of covariates that correlates poorly with the criterion exacerbates the degree of bias as follows. As Cochran (1957) and Elashoff (1969) have shown, the increase in precision of the F-test for main effects can be expressed in terms of the decrease in error variability due to the addition of a covariate. This decrease can be expressed as a function of the correlation between the criterion and the covariate. The decreased error variability is

$$\sigma_{e.X}^2 = \sigma_e^2 (1 - \rho^2) (1 + 1/(f_e - 2)) \qquad (8)$$

where $\sigma_{e.X}^2$ is the error variability with the covariate added to the model, σ_e^2 is the error variability as obtained in the ANOVA model, ρ is the correlation between the criterion and the covariate, and f_e are the degrees of freedom associated with error term. This equation can be rearranged to yield a formula for the proportionate reduction in error variability due to addition of a covariate. The equation becomes

$$\Delta_e/\sigma_e^2 = (\sigma_e^2 - \sigma_{e.X}^2)/\sigma_e^2 = [\rho^2 (f_e - 1) - 1]/(f_e - 2) \qquad (9)$$

The reduction results in an unbiased parameter estimate only when the correlation between covariate and grouping variables is zero. It is the case that even though the population coefficient may be zero, the sample coefficient deviates slightly from zero. In that case, the independence of the covariate as a predictor of the criterion is represented by $\rho_{eX.G}$, the part correlation reflected in the regression weight of the covariate. In this the independent variable and criterion compete for the covariate variance. Since the redundant variance has been removed from the error term by inclusion of the independent variable, the proportionate reduction in error variability decreases as the assumption of independence is violated.

The change in error variability can be estimated. For n=100 and a correlation of .20 between the criterion and covariate (a correlation just above the .05 level of significance), one reduces the error variance by about 3%. With a sample of 100 observations, it is conceivable that the correlation of .20 reflects a chance relationship. If this is the case, adjustment to the analysis of variance is performed for a variable that is not a member of the set $\{X_i, \ldots, X_i\}$ for the function in Equation 2. This adjustment thus is tantamount to arbitrarily pulling error variation out the denominator of the F-test.

Weisberg (1979) has demonstrated for the uncontrolled study in the case of the linear model that the proportion of bias remaining after the ANCOVA adjustment is a function of three correlations: (1) ρ_{ZO}, the correlation between group membership and a criterion variable calculated under the imaginary condition that all individuals were assigned to the control condition (or in the case of an individual difference variable, to the same value of the variable); (2) ρ_{ZX}, the correlation between that same criterion measure and the covariate; and (3) ρ_{XO}, the correlation between the covariate and group membership. The proportion of bias remaining after adjustment

(Weisberg, 1979) can be expressed as

$$\pi = \frac{\rho_{ZQ} - \rho_{ZX}\rho_{XQ}}{\rho_{ZQ}(1 - \rho_{XQ}^2)} \qquad (10)$$

In practice, only ρ_{XQ} can be estimated from the data. However, it can be shown that the covariate adjustment is unbiased only for the cases in which $\rho_{ZX} = 1$. For our purposes, this equation serves to demonstrate that even in situations in which one is able to determine precision in the analysis, one can at best only estimate bias for the uncontrolled study. The next section presents data which indicate degree of that bias.

Bias Due to Low Covariate-Criterion Correlations

The following discussion concerns the situation in which a non-zero but non-significant correlation between the covariate and the criterion exists. The cause of such a correlation could be, in part, due to the lack of reliability of the covariate. However, the situation with which this paper is concerned is that in which a variable assumed to be covariate has a true correlation of zero with the criterion variable.

Two reports of simulation studies will be presented. The first will concentrate on one covariate in the context of a one-factorial design. The second will present two or more covariates in a two-factorial design.

Study I: Method and Results

To show the degree of bias introduced when controlling for a statistically non-significant relationship, a simulation study and analysis of empirical data were conducted in which a set of one-way ANOVAs on a criterion variable were performed. The grouping variable used had only two levels. The criterion variable was created by generating a random normal variate and assigning a grouping number (either 1 or 2) to each value of the variate. A constant was then added to the second group creating the group difference. The size of the constant was initialized at d = .02 and incrementing by .05 until the difference became statistically significant at the $p < .001$ level. Covariates were selected by generating a set of random variates (also selected from a normal distribution) using the RANNOR function in SAS and correlating each variate with the criterion measure. The selected covariates had correlations with the criterion ranging from r = .01 up to a level of correlation representing a relationship just under the alpha = .05 level of significance. Covariates were assumed to be homogeneous across and independent of groups. Models representing each level of difference were then re-estimated including the covariate. The simulation was performed for n = 100. The results of the analysis appear in Table 1.

The table shows the patterns of the tail probabilities of the significance tests of the estimated model. The change in tail probabilities from the ANOVA F-test to the ANCOVA F-test is a function of the correlation between criterion and covariate as one would expect. For the case in which the correlation is near zero, and the adjustment is small compared to the sampling fluctuation, the change in tail probabilities tend to oscillate about the nominal alpha level. However, within each a sufficiently high correlation causes adjustments in the direction of a more liberal test. The size of the correlation sufficient to effect a change is surprisingly small. For n = 100 and a mean difference of .050 between the two simulation groups, the tail probability of the F-test is .0721. A correlation between the covariate and criterion of r = .17 ($p = .0967$) changes the tail probability to .0463 which becomes statistically significant if one uses the $p < .05$ cutoff.

Discussion

The degree of bias introduced into an analysis-of-variance F-test by the inclusion of a covariate that is weakly correlated with the criterion variable depends on both the size of the correlation and the size of the sample. The degree of the bias can roughly be estimated by calculating the reduction in error variance in Equation 5 using the largest possible nonsignificant correlation between the covariate and the criterion. Using this equation as an estimate assumes that the reduction in variance identified as the increase in precision functions as a measure of bias. This is the case only when the covariate-criterion relationship can be attributed only to sampling fluctuation. The results of this study suggest that when the correlation is non-zero and nonsignificant the tail probability of the test will most often be underestimated. In the case of a marginally non-significant test, bias (even with an n as large as 100) can be enough to push the probability value into the significant area of the sampling distribution.

The decision whether to include a covariate as a statistical control in an analysis should be based on a number of criteria; the most important of which is the degree to which the adjustment represented by the covariate is theoretically meaningful. Once that criterion is met, one should determine an absolute minimum value of the criterion-covariate relationship. The minimum value will be a function of the size of the correlation, the sample size, and the size of the effect. It is suggested here that an absolute minimum be determined by requiring the correlation to be above a level of significance determined by a priori for the study and adjusted for the number of covariates considered.

While the models discussed above included only one covariate, one can imagine the situation with more than one covariate. Assuming two uncorrelated covariates, each would tend to decrease the error term by some amount that would tend to decrease the size of the tail-probability even more. One could essentially control the level of significance by introducing enough weak covariates into any analysis.

To demonstrate this, we ran the following simulation study to show the effect of selecting

Table 1

The Effect of a Random Covariate on a One-Way ANOVA (n=100)

(Tail Probabilities of the F-Test)

	Mean Differences Between Groups												
Correlation	.020	.025	.030	.035	.040	.045	.050	.055	.060	.065	.070	.075	.080
No Covariate	.8281	.6291	.4543	.3112	.2019	.1240	.0721	.0397	.0207	.0102	.0048	.0022	.0009
.01 $\underline{p}$ = .9358	.8292	.6311	.4568	.3138	.2043	.1260	.0736	.0407	.0214	.0106	.0050	.0038	.0010
.03 $\underline{p}$ = .7538	.8137	.6174	.4455	.3051	.1981	.1218	.0710	.0392	.0205	.0102	.0048	.0022	.0009
.05 $\underline{p}$ = .6214	.7716	.5817	.4175	.2846	.1842	.1130	.0658	.0364	.0191	.0095	.0045	.0020	.0009
.07 $\underline{p}$ = .4730	.7644	.5737	.4094	.2773	.1781	.1084	.0625	.0342	.0177	.0087	.0041	.0018	.0008
.09 $\underline{p}$ = .3654	.8333	.6341	.4586	.3147	.2046	.1259	.0734	.0405	.0212	.0105	.0049	.0022	.0010
.11 $\underline{p}$ = .2612	.7048	.5212	.3662	.2442	.1543	.0924	.0524	.0282	.0144	.0070	.0032	.0014	.0006
.13 $\underline{p}$ = .1620	.7443	.5537	.3911	.2619	.1660	.0996	.0565	.0304	.0155	.0075	.0034	.0015	.0006
.15 $\underline{p}$ = .1233	.7575	.5648	.3997	.2681	.1701	.1021	.0580	.0312	.0159	.0077	.0035	.0015	.0006
.17 $\underline{p}$ = .0967	.6781	.4960	.3442	.2263	.1409	.0830	.0463	.0244	.0122	.0058	.0026	.0011	.0005
.19 $\underline{p}$ = .0572	.6882	.5035	.3493	.2294	.1425	.0837	.0465	.0245	.0122	.0058	.0026	.0011	.0005

Table 2

Analyses in Which the "Nonsignificant Covariates" are Retained

	Null Hypothesis .05 (.01)	Alternate Hypothesis .05 (.01)
Overall Model	14% (4%)	100% (96%)
Independent Variable A	25% (8%)	99% (99%)
Independent Variable B	17% (9%)	98% (96%)

a set of "significant" covariates generated from a set of random variates.

Study II: Method and Results

Monte Carlo data for a series of 100 double classification analyses of covariance were generated using random data. For these simulations, care was taken to approximate the sample sizes and effect sizes found in developmental psychology research. In the analysis of covariance, two main effects were defined with three levels each, and 16 normal covariates. Each cell of the analysis contained 12 observations, making a total sample size of 108 for each analysis. All data were generated with SAS random number functions for generating normal variates (function RANNOR). For the first series of 100 "experiments" no main effect or interaction was defined in the dependent variable.

Analyses Based on Data Where the Null Hypothesis is True

For the first set of Monte Carlo data where the null hypothesis was true, an analysis of covariance was conducted to estimate the effect of false inclusion of covariates on the nominal alpha level of the experiment. Any covariate which achieved statistical significance at the .05 level was left in the model for the data. This situation would correspond to that arising when the researcher falsely concludes that a covariate is statistically significant when it is, in fact, not. Comparison of the probability levels associated with the overall model, and with the effects of the two independent variables form the basis for comparing the nominal error rates of the model with the true error rates. For probability levels associated with the overall model, 14% of the demonstrated significance at a nominal .05 alpha level. Four percent of the experiments demonstrated a nominal alpha level of .01. For the independent variables, 25% and 17% of the experiments showed a nominal alpha level of .05 for the first variable and second independent, respectively. The proportions of experiments showing a nominal alpha of .01 was 8% and 9%, respectively.

Discussion

As can be seen, selection of those "covariates" that appear to be significant and adjustment for those variables can dramatically change the tail probabilities of the F-test. In practice such an error of inclusion can be avoided by carefully controlling the experiment-wise alpha level used to define a significant correlation. However, researchers looking at single covariates that account for what appears to be a small part of the variation in a criterion variable are often reluctant to give up that adjustment. By keeping such variables as covariates, they must realize that ANCOVA adjusts for anything that is supplied as a covariate in the analysis. ANCOVA will indeed attempt to adjust even when the covariate-criterion relationship is not more than chance. Since the proportion of bias remaining (Equation 11) after the adjustment is inversely proportional to the size of the covariate-criterion correlation, care must be taken in the selection of variables that act as covariates.

It is ultimately the choice of the investigator to determine which variable might theoretically serve as covariates. However, as Weisberg (1979) pointed out, inclusion is only appropriate if one can assume that individuals who have the same value on a covariate, and are members of different groups, would have the same value on the criterion in the absence of a group effect. We add that even when this assumption is plausible, one must be assured that the size of the covariate-criterion relationship signifies a truly non-zero and theoretically meaningful relationship.

References

Belsky, J., Gilstrap, B., & Rovine, M. (1984). The Pennsylvania Infant and Family Development Project, I: Stability and change in mother-infant and father-infant interaction in a family setting at one, three, and nine months. Child Development, 55, 692-705.

Cochran, W. G. (1957). Analysis of covariance: Its nature and uses. Biometrics, 13, 261-281.

Cohen, J., & Cohen, P. (1975). Applied multiple regression/correlation analysis for the behavioral sciences. Hillsdale, NJ: Erlbaum.

Cook, T. D., & Campbell, D. T. (1979). Quasi-experimentation. Chicago: Rand McNally.

Cox, D. R. (1957). The use of a concomitant variable in selecting an experimental design. Biometrika, 44, 150-158.

Elashoff, J. D. (1969). Analysis of covariance: A delicate instrument. American Educational Research Journal, 6, 383-401.

Fisher, R. A. (1932). Statistical methods for research workers. Edinburgh, Scotland: Oliver and Boyd.

Glass, G. V., Peckham, P. D., & Sanders, J. R. (1973). Consequences of failure to meet assumptions underlying the fixed effects analysis of variance and covariance. Review of Educational Research, 42, 237-288.

Maxwell, S., Delaney, H., & Dill, C. (1984). Another look at ANCOVA versus blocking. Psychological Bulletin, 95, 136-147.

Myers, J. L. (1979). Fundamentals of experimental design. Boston: Allyn and Bacon.

Weisberg, H. I. (1979). Statistical adjustments and uncontrolled studies. Psychological Bulletin, 86, 1149-1164.

Estimation of the Variance Matrix for Maximum Likelihood Parameters Using Quasi-Newton Methods

Linda Williams Pickle[1] , National Cancer Institute
Garth P. McCormick, George Washington University

Abstract

With recent advances in computer processing speed, statistical packages with generalized maximum likelihood estimation subroutines are proliferating. Unfortunately, convergence criteria in these packages are based on the step-wise change of the parameter estimates or on the closeness of the first derivative vector to 0. No measure of the adequacy of the asymptotic parameter variance matrix exists and most statisticians are unaware that the variance matrix approximation based on the commonly-used quasi-Newton iterative methods can be poor. We examine the behavior of this approximation for two representative likelihoods and suggest an additional convergence criterion that may help the user to determine when the variance matrix as well as the parameter vector are sufficiently close to their true values.

1. Introduction

Maximum likelihood parameter estimation requires an iterative solution for all but the simplest statistical distributions for which analytic solutions are available. The most popular method of solution has been Newton's method, which converges quickly and automatically provides the most accurate estimate available of the asymptotic variance matrix for the parameter estimates. Unfortunately, this method requires the user to provide the second derivatives of the likelihood function (the Hessian matrix), a task which is often very difficult. The quasi-Newton class of unconstrained optimization algorithms avoids this problem by approximating the inverse Hessian matrix, and has been shown to converge superlinearly to the correct parameter vector. Use of a quasi-Newton algorithm and an accurate first derivative approximation algorithm coupled with the recent dramatic improvements in microcomputer processing speed has made possible generalized computer programs for maximum likelihood estimation. It is no longer necessary for the statistician to write a new FORTRAN program for each different form of the likelihood to be maximized, thus simplifying examination of competing models.

Because the quasi-Newton methods were developed for the solution of deterministic models, primarily in operations research, little work has been done to examine the accuracy of the approximation for the variance matrix or its rate of convergence. We have examined the behavior of this matrix approximation for several representative likelihoods. Comparison of known analytic results to results from a quasi-Newton procedure using an optimal step size suggests that after the first few iterations the variance matrix approximation converges to its correct values at nearly the same rate as the parameter vector itself. We propose a method of determining when the matrix has converged sufficiently to a solution.

2. Maximum Likelihood Estimation

Let Z_r denote the data vector for observation r and $X = (x_1, x_2, \cdots, x_J)'$ denote the parameter vector to be estimated. Then

$$L(Z;X) = \prod_{r=1}^{N} f(Z_r;X), \tag{1}$$

is the likelihood function to be maximized over possible values of X, where $f(Z_r;X)$ is the probability density function for X. Maximum likelihood estimation may be viewed as a general unconstrained optimization problem, requiring solution of any of the following equivalent problems:

$$\max_X L(Z;X) \text{ or } \min_X[-L(Z;X)] \text{ or } \min_X[-\ln L(Z;X)]$$

For convenience, we will solve the last problem. Let $G(Z;X) = -\ln L(Z;X)$, the objective function to be minimized. If we may assume that:

1. $G(Z;X)$ is twice continuously differentiable,
2. X^* is the unique solution to the problem; X^* is called the maximum likelihood estimate of X,
3. $G(Z;X)$ is strictly convex in a neighborhood about X^* (i.e., $L(Z;X)$ is strictly concave),

then

$$E\left(\frac{\partial G(Z;X)}{\partial X}\right)\Big|_{X^*} = 0 \tag{2}$$

and

$$E\left(\frac{\partial^2 G(Z;X)}{\partial X_j \partial X_k}\right)^{-1} = \mathrm{Var}(X), \tag{3}$$

the asymptotic variance matrix of the parameter vector

[1] *current address:* Vincent T. Lombardi Cancer Research Center, Georgetown University, Washington, DC 20007

X. In practice, we estimate this variance matrix by H^*, the inverse Hessian matrix evaluated at the maximum likelihood estimates of the parameters (X^*).

3. Iterative Methods

For all but the simplest models, analytic solutions to the maximum likelihood optimization problem do not exist. Iterative methods of solution generally include the following steps:

1. determine initial estimates for X, denoted X_0
2. compute $X_{i+1} = X_i + S_i t_i$
   where $X_i = (x_{i1}, x_{i2}, \cdots, x_{iJ})'$, the parameter vector estimate at iteration i; S_i is the direction vector for this step and t_i is the step size which minimizes G along the ray S.
3. repeat step 2 until convergence or the maximum number of iterations is reached.

For the Newton method of solution,

$$S_i = - \left(\frac{\partial^2 G(Z;X)}{\partial X_j \partial X_k} \right)^{-1} \left(\frac{\partial G(Z;X)}{\partial X} \right) \Bigg|_{X_i} . \quad (4)$$

Calculation of the optimal step size t_i is required to ensure a significant improvement at each step (Chambers 1977). However, most implementations of Newton's algorithm use a constant step size of 1, which is optimal for a quadratic objective function, and only adjust t_i (e.g., by step halving) when no improvement is seen in the solution vector.

The Newton method converges quadratically to the solution vector but requires the first and second partial derivatives of the objective function and inversion of the Hessian matrix at each step. Because the second partial derivatives are often tedious to calculate, quasi-Newton methods approximate the inverse Hessian by H_i, an update of the last iterate's matrix; that is, $H_{i+1} = H_i + \delta_i$. Then

$$S_i = - H_i \left(\frac{\partial G(Z;X)}{\partial X} \right) \Bigg|_{X_i} . \quad (5)$$

Several commonly used updating methods are the Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) methods (McCormick 1983, chap. 9).

Obvious advantages of the quasi-Newton methods are that no second derivatives or matrix inversion are required. By combining a quasi-Newton inverse Hessian update algorithm with algorithms to approximate the first derivative vector (i.e., the score vector) and to calculate the optimal step size at each iteration, a generalized optimization program may be developed that requires the user to provide only the objective function.

Unfortunately, these inverse Hessian approximations can be poor estimates of the asymptotic variance matrix. This is a critical defect for optimization of statistical problems, since the variance matrix is used to test the significance of the parameter estimates and to calculate confidence limits for the parameters.

The order of the rate of convergence of an iterative procedure is defined to be the power p such that

$$\lim_{i \to \infty} \frac{\|X_{i+1} - X^*\|}{\|X_i - X^*\|^p} = M \quad (6)$$

where M is a constant. $\|C\|$ denotes a norm of any vector C; we will use the L_2 norm; i.e., $\|C\| = (\sum_j c_j^2)^{1/2}$. An algorithm is said to converge superlinearly if $p > 1$. It may be shown that Newton's method converges quadratically (i.e., p=2). For secant methods in general, a class of algorithms including quasi-Newton methods, Tornheim (1964) showed that the asymptotic order of the rate of convergence is the solution to the equation $p^{J+1} - p^J - 1 = 0$, where J is the number of parameters to be estimated. For example, this result shows that for a single parameter $p = (1 + \sqrt{5})/2 = 1.618$. The rate of convergence is slower with an increasing number of parameters, but in all cases is superlinear.

In practice, we may estimate the order of the rate of convergence of the parameter estimates by taking logarithms of both sides of (6), and solving the following simple linear regression problem for p over successive iterations ($i = 1, 2, \cdots, I$):

$$\ln \|X_{i+1} - X^*\| \quad \alpha + p \ln \|X_i - X^*\|. \quad (7)$$

Obviously, if we are considering using the quasi-Newton method at all, an analytic solution is unavailable. Thus, the asymptotic order of the rate of convergence only provides a guide to the rate of convergence expected in practice, and is not useful in determining whether convergence has been reached in any particular situation. How can we judge whether the H matrix approximation adequately represents the true variance matrix?

Let $\sigma_i = X_{i+1} - X_i$, the difference in the parameter estimates from one iteration to the next, and $y_i = \nabla G(Z; X_{i+1}) - \nabla G(Z; X_i)$, the difference in the first derivatives of the objective function from one iteration to the next. Typical quasi-Newton methods update the H matrix (i.e., $H_{i+1} = H_i + \delta_i$) so that $H_{i+1} y_i = \sigma_i$. It seems reasonable to expect that as $H_m \to H^*$, the correct inverse Hessian matrix, H_m will solve several of the previous equations as well. That is,

$$\begin{aligned} H_m y_{m-2} &\approx \sigma_{m-2}, \\ H_m y_{m-3} &\approx \sigma_{m-3}, \\ &\vdots \\ H_m y_{m-J+1} &\approx \sigma_{m-J+1} \end{aligned}$$

where J is (arbitrarily) the number of parameters to be estimated. If this is true, then we may use $\|H_m y_i - \sigma_i\| / \|\sigma_i\|$ to measure the adequacy of the H approximation.

4. Computational Methods

The BFGS quasi-Newton algorithm was implemented as a subroutine of a program that calculates analytic derivatives and includes an optimal step size routine. This program also calculated the norm measures for each iteration as described above. All programs were written in Fortran-77 and were run on an IBM 3090 system at the National Institutes of Health. All calculations were performed in double precision. Machine epsilon for this system is approximately 10^{-15}.

The theoretical order of the rate of convergence was calculated by solving Tornheim's equation for the appropriate number of parameters (Borland International, Inc. 1987). Although these theoretical results apply to the convergence of the parameter vector, we used the same techniques to examine the convergence behavior of the inverse Hessian approximation (H). The observed order was calculated using the linear regression function of Lotus 1-2-3, version 2.01 (Lotus Development Corp. 1985). However, because a sharp improvement in H is expected on the Jth step when sufficient information is available to approximate the JxJ matrix (McCormick 1983, p.198), we omitted this step from the regression data.

5. Results

5.1. Example 1: Logistic model

Data for the first example are from a case-control study of laryngeal cancer among white male residents of the Texas Gulf Coast area (Brown 1988). A total of 209 cases and 250 controls (or their next of kin) were successfully interviewed to obtain information on their usual consumption of alcohol and tobacco, as well as information on other potential risk factors for this tumor. A prospective logistic model was used to estimate the relative risk for laryngeal cancer due to joint exposure to alcohol and tobacco, adjusting for age. The likelihood to be maximized is:

$$L(Z;X) = \prod_{r=1}^{N} \frac{[\exp(X_0 + X_1Z_{1r} + X_2Z_{2r} + X_3Z_{3r})]^{d_r}}{1 + \exp(X_0 + X_1Z_{1r} + X_2Z_{2r} + X_3Z_{3r})} \quad (8)$$

where

$$d_r = \begin{cases} 1 & \text{if person } r \text{ had laryngeal cancer} \\ 0 & \text{if not} \end{cases}$$

Z_{1r} = # packs smoked per day by person r

$$Z_{2r} = \begin{cases} 1 & \text{if person } r \text{ was a heavy alcohol drinker} \\ 0 & \text{if not} \end{cases}$$

$$Z_{3r} = \begin{cases} 1 & \text{if person } r \text{ was age 60+} \\ 0 & \text{if not} \end{cases}$$

$X = (x_0, x_1, x_2, x_3)'$ are the parameters to be estimated, where X_j is the parameter corresponding to each $Z_j, j = 1,2,3$, and x_0 is a constant.

Results from a program using Newton's method with exact first and second derivatives (Harrell 1986) show that the maximum likelihood parameter estimates are $X^* = (-2.81, 0.81, 1.26, 0.36)'$. As shown in Figure 1, X_i converged quickly to these values; after 6 iterations the norm of the relative error was 0.0001. H converged to its correct values (H^*) less quickly; as expected, there was a dramatic improvement on the fourth step. After this improvement in H the vector of first derivatives (y) dropped rapidly to 0.

Results of the linear regression procedure estimated the order of convergence of X_i to be 1.20, compared to the predicted asymptotic order of 1.33. A similar calculation for H_i, excluding step 4, estimated an order of convergence of 1.06.

In practice, when the correct results are unavailable, a commonly used stopping rule (i.e., convergence criterion) is to require a "small" value of the maximum relative parameter change from one iteration to the next. That is, for step i

$$\max_j \left| \frac{x_{ij} - x_{i-1,j}}{x_{i-1,j}} \right| < \epsilon \quad (9)$$

where ϵ is a small positive number and j indexes the parameters in the vector X.

For this example, as shown in Figure 2, we might choose to stop after iteration 7, where the maximum relative parameter change was less than 0.001.

It is less clear from Figure 2 when H has improved sufficiently to declare convergence. In fact, this measure of convergence did not decrease monotonically over the 11 iterations shown. As described previously, we calculated $\|H_m y_i - \sigma_i\| / \|\sigma_i\|$ for each $i < m - 1$. Figure 3 shows the results for m=7,8,9 and 10. Unlike the conventional convergence criteria described above, we do see an improvement with each iteration; for this example H_8 and H_9 both satisfy the previous 4 equations to within a tolerance of 0.02.

5.2. Example 2: Normal mixture model

Because the logistic example converged in so few iterations, we repeated the maximization on a more complex likelihood function. Data were simulated for a mixture of normal distributions, with 400 points generated from a N(0,1) distribution and 100 from a N(4,1) distribution (SAS Institute, Inc. 1985). The likelihood to be maximized was:

$$L(Z;X) = \prod_{r=1}^{N} \frac{1}{\sqrt{2\pi}} \left\{ \frac{p}{\sigma_1} \exp\left[\frac{-(Z_r - \mu_1)^2}{2\sigma_1^2}\right] + \frac{(1-p)}{\sigma_2} \exp\left[\frac{-(Z_r - \mu_2)^2}{2\sigma_2^2}\right] \right\} \quad (10)$$

where Z_r is the observed data vector for person r, μ_1 and μ_2 are the means and σ_1 and σ_2 the standard deviations

for the two normal distributions, p is the proportion of the total in distribution 1, and $X = (p, \mu_1, \sigma_1, \mu_2, \sigma_2)'$. We assume $\mu_1 < \mu_2$ so that the solution is unique.

Using a Newton method with exact derivatives we calculated the true maximum likelihood parameter estimates to be $X^* = (0.811, -0.0315, 1.02, 4.11, 0.900)'$. As shown in Figure 4, the patterns of convergence to the true solution follow those seen in the logistic example. After 10 iterations the norm of the relative error of the parameter vector was approximately 0.0001. Following a sharp improvement on step 5, H improved steadily and y converged to 0.

The order of the rate of convergence for the parameter vector was 1.17, compared to an asymptotic predicted order of 1.28 for 5 parameters. A similar calculation for H, again excluding the sharp drop on step 5, shows the order of convergence to be 0.98.

Use of the maximum relative parameter change to define convergence would lead to stopping after iteration 12 for $\epsilon = 0.001$ (Figure 5). As with the logistic model, this measure does not monotonically decrease for H. Application of each H_i to previous iterations showed steady improvement with each iteration (Figure 6). H_{17} solves the previous 5 iteration equations within a tolerance of 0.02.

6. Conclusions

For two representative maximum likelihood estimation problems, the order of the rate of convergence of the parameter vector to the correct solution was nearly equal to its predicted asymptotic value. The inverse Hessian approximation, the estimated asymptotic variance matrix for the parameters, showed an order of convergence slightly less than that of the corresponding parameter vector, but converged at least linearly in both examples. The proposed norm measure of the closeness of the inverse Hessian matrix approximation to its correct values appears promising, showing an improvement at each iteration and agreeing with our decisions to declare convergence when the true matrix values are known. Addition of a criterion such as this to the standard stopping rules should increase the number of iterations performed to ensure that the variance matrix as well as the parameter vector is reasonably close to the true values or identify situations where the values are suspect. Although the examples used here seem sufficiently difficult nonlinear optimization problems, a wider range of likelihoods needs to be examined. We encourage continued work in this area so that statisticians may use the newly available computer methods with confidence.

7. References

Borland International, Inc. (1987), *Eureka: The Solver*, Scotts Valley, CA: Author.

Brown, L. M., Mason, T. J., Pickle, L. W., Stewart, P. A., Buffler, P. A., Burau, K., Ziegler, R. G., and Fraumeni, J. F., Jr. (1988), "Occupational Risk Factors for Laryngeal Cancer on the Texas Gulf Coast," *Cancer Research*, 48, 1960-1964.

Chambers, J. M. (1977), *Computational Methods for Data Analysis*, New York, N.Y.: John Wiley and Sons, pp. 136-145.

Harrell, F. E., Jr. (1986), "The LOGIST Procedure," in *SUGI Supplemental Library User's Guide, 1986 Edition*, Cary, N.C.: SAS Institute, Inc., pp. 269-293.

Lotus Development Corp. (1985), *1-2-3 Reference Manual*, Cambridge, MA: Author.

McCormick, G. P. (1983), *Nonlinear Programming*, New York, N.Y.: John Wiley and Sons.

Tornheim, L. (1964), "Convergence of Multipoint Iterative Methods," *Journal of the ACM*, 11, 210-220.

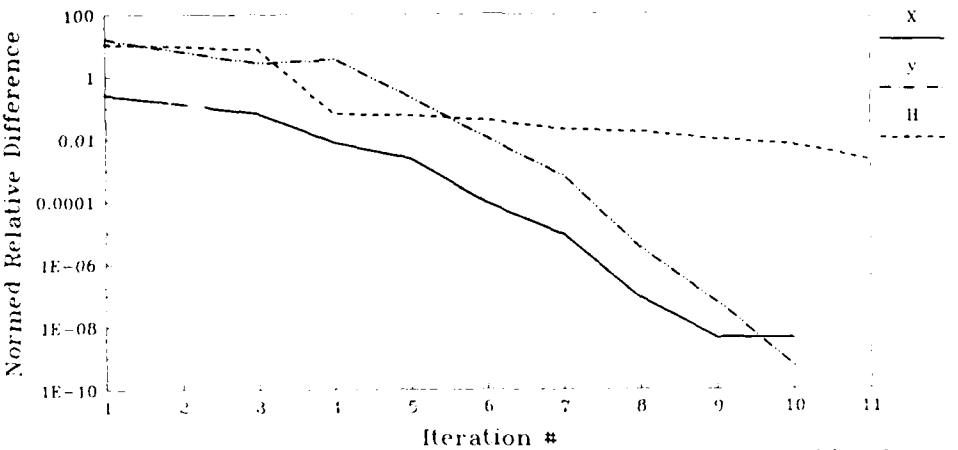


Figure 1. Convergence of Estimates of Parameters (X), First Derivatives (y), and Inverse Hessian Matrix (H) to True Values for Logistic Model Example. Normed relative difference = $\|X_i - X^*\|/\|X^*\|$ for parameter vector X_i estimated at iteration i, when true parameter values = X^*; similar definitions for y and H.

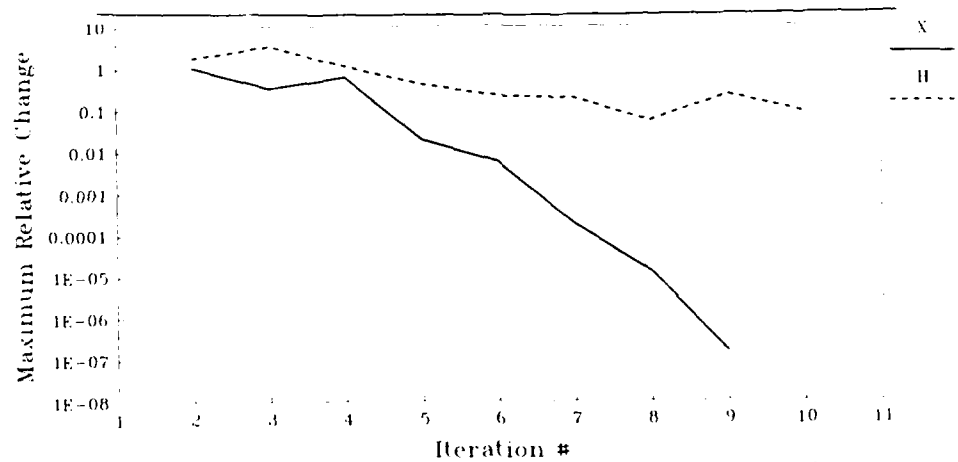


Figure 2. Maximum Relative Change of Estimates of Parameters (X) and Inverse Hessian Matrix (H) for Logistic Model Example. Maximum relative change from iteration $i-1$ to $i = \max_j \left| \frac{X_{ij} - X_{i-1,j}}{X_{i-1,j}} \right|$ for X; similar for H.

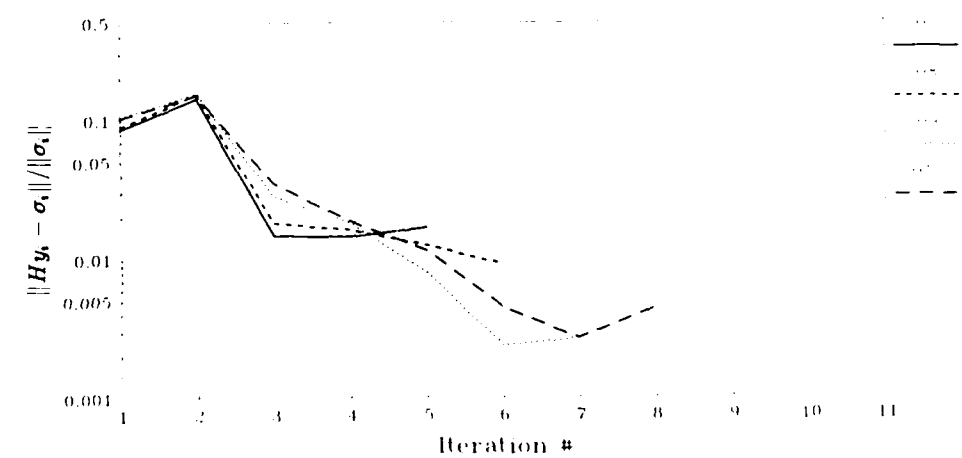


Figure 3. Solution of the Equation $Hy = \sigma$ by the Inverse Hessian Approximations from Iterations 7, 8, 9, and 10 using y and σ from Previous Iterations; Logistic Model Example.

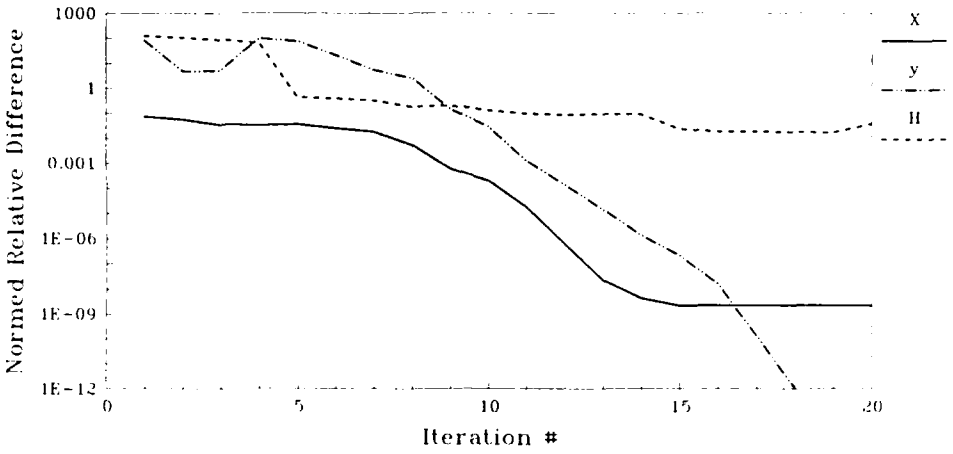


Figure 4. Convergence of Estimates of Parameters (X), First Derivatives (y), and Inverse Hessian Matrix (H) to True Values for Normal Model Example. Normed relative difference = $\|X_i - X^*\|/\|X^*\|$ for parameter vector X_i estimated at iteration i, when true parameter values = X^*; similar definitions for y and H.

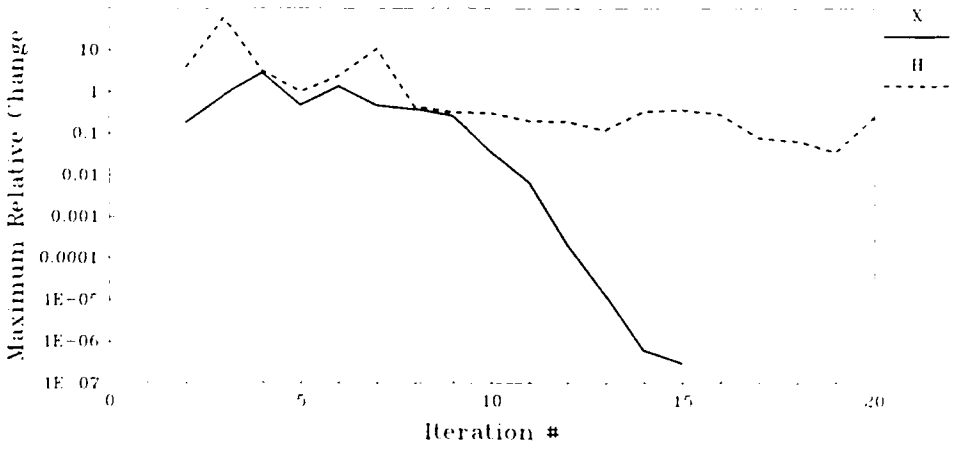


Figure 5. Maximum Relative Change of Estimates of Parameters (X) and Inverse Hessian Matrix (H) for Normal Model Example. Maximum relative change from iteration $i-1$ to $i = \max_j \left| \frac{X_{ij} - X_{i-1,j}}{X_{i-1,j}} \right|$ for X; similar for H.

Figure 6. Solution of the Equation $Hy = \sigma$ by the Inverse Hessian Approximations from Iterations 15, 17, and 19 using y and σ from Previous Iterations; Normal Model Example.

APPLICATION OF POSTERIOR APPROXIMATION TECHNIQUES TO THE ORDERED DIRICHLET DISTRIBUTION

Thomas A. Mazzuchi and Refik Soyer

The George Washington University

1. INTRODUCTION AND OVERVIEW

The ordered Dirichlet distribution has proved to be a meaningful prior distribution in a variety of applications including bioassay [Ramsey (1972)], life testing [Lochner (1975)], damage response [Mazzuchi and Singpurwalla (1982)], failure rate estimation [Mazzuchi and Singpurwalla (1983)], and accelerated life testing [Mazzuchi (1986)]. In the above situations the distribution is used as a prior distribution for a set of ordered probabilities. There are three reasons that the ordered Dirichlet distribution is so appealing,

i. It imposes no other restriction on the probabilities other than the desired ordering.
ii. It allows for easy incorporation of prior information.
iii. It is mathematically tractable and allows for closed form posterior results.

While the first two claims are valid, the third is only partially true. It is possible to obtain results in closed form, however, evaluation of some posterior quantities may require large amounts of computer time and may be subject to error due to numerical manipulations. These errors are a function of the sample size and tend to increase as the sample size increases. With larger sample sizes we therefore offer as an alternative the use of the posterior approximation technique developed by Tierney and Kadane (1986) for obtaining posterior quantities.

In Section 2 we present an overview of the use of the ordered Dirichlet distribution with both prior and posterior results. In Section 3 we present an overview of the Tierney Kadane method and show the ease with which this method can be used to obtain the posterior quantities of section 2. In Section 4 we give some closing comments.

2. THE ORDERED DIRICHLET DISTRIBUTION

The ordered Dirichlet distribution defined for a set of variables $\underset{\sim}{u} = (u_1, u_2, \ldots, u_k)$ is given by

$$f(\underset{\sim}{u} \mid \beta, \underset{\sim}{\alpha}) = \frac{\Gamma(\beta)}{\prod_{j=1}^{k} \Gamma(\beta\alpha_j)} \prod_{j=1}^{k+1} (u_{j-1} - u_j)^{\beta\alpha_j - 1} \qquad (2.1)$$

with

$\beta > 0$,

$\alpha_j > 0$, $j = 1, \ldots, k+1$, and

$\sum_{j=1}^{k+1} \alpha_j = 1$.

The joint distribution is defined over the simplex

$$1 = u_0 \geq u_1 \geq u_2 \geq \cdots \geq u_k \geq u_{k+1} = 0,$$

thus preserving the desired ordering. It is easy to see that this distribution arises as a result of specifying a Dirichlet distribution on the successive forward differences of the above variables.

2.1 Prior Results

Prior information may be directly incorporated through the prior parameters by noting that

$$E[u_i] = \sum_{j=i+1}^{k+1} \alpha_j \qquad (2.2a)$$

$$Var[u_i] = \frac{E[u_i] * \left(1 - E[u_i]\right)}{\beta+1} \qquad (2.2b)$$

and thus if $u_1^*, \ldots, u_k^*$ are the prior best guess values for $u_1, \ldots, u_k$, defining $\alpha_i = u_{i-1}^* - u_i^*$, $i = 1, \ldots, k+1$ (with $u_0^* = 1$ and $u_{k+1}^* = 0$) we obtain a joint distribution whose marginal mean values are our prior best guesses. In addition, the parameter β may be specified in such a way as to indicate the strength of conviction in these prior best guess values. This is true since once the α_i are specified, β controls the magnitude of the variance.

2.2 Posterior Results

Without getting into specific problem scenarios, the general form for the likelihood in problems using the ordered Dirichlet distribution is given by

$$\mathcal{L}(\underset{\sim}{n}, \underset{\sim}{s}; \underset{\sim}{u}) \propto \prod_{j=1}^{k} (1 - u_j)^{s_j} u_j^{n_j - s_j} \qquad (2.3)$$

where n_j and s_j are quantities used for estimating u_j (essentially n_j indicates the sample size for estimating u_j and s_j is often the number of failures recorded out of n_j). The posterior joint distribution for $\underset{\sim}{u}$ is thus proportional to the product of (2.1) and (2.3) and this is given by

$$\prod_{j=1}^{k} (1-u_j)^{s_j} u_j^{n_j - s_j} (u_{j-1} - u_j)^{\beta\alpha_j - 1} u_k^{\beta\alpha_{k+1}-1} \tag{2.4}$$

We can expand the $(1-u_j)^{s_j}$ terms in a binomial series yielding

$$\sum_{\ell_1=0}^{s_1} \cdots \sum_{\ell_k=0}^{s_k} \prod_{j=1}^{k} \left\{ \binom{s_j}{\ell_j} (-1)^{\ell_j} u_j^{\ell_j + n_j - s_j} (u_{j-1} - u_j)^{\beta\alpha_j - 1} \right\} u_k^{\beta\alpha_{k+1}-1} \tag{2.5}$$

This can be expressed as a weighted combination of densities of a form proportional to

$$\prod_{j=1}^{k} u_j^{a_j} (u_{j-1} - u_j)^{b_j - 1} u_k^{b_{k+1}-1} \tag{2.6}$$

The above density is similar to the generalized Dirichlet density studied by Lochner(1975) and Connor and Mosimann (1969). The constant of integration and thus the moments for this distribution may easily be obtained through repeated use of the integral identity (CRC TABLES Definite Integral Formula 609)

$$\int_a^b (x-a)^m (b-x)^n \, dx = (b-a)^{m+n+1} B(m+1, n+1)$$

where $B(m+1,n+1) = \dfrac{\Gamma(m+1)\,\Gamma(n+1)}{\Gamma(m+n+2)}$ is the beta function. Thus the constant of integration for (2.6) is obtained as

$$\prod_{j=1}^{k} B\left(\sum_{m=j}^{k} a_m + b_{m+1}, b_j \right) \tag{2.7}$$

and joint moments $E[u_c^{\nu_1} u_d^{\nu_2} | \underset{\sim}{a}, \underset{\sim}{b}]$ for (2.6) are obtained as

$$\left[\prod_{j=1}^{c} B\left(\nu_1 + \nu_2 + \sum_{m=j}^{k} a_m + b_{m+1}, b_j \right) \Big/ \prod_{j=1}^{c} B\left(\sum_{m=j}^{k} a_m + b_{m+1}, b_j \right) \right] \left[\prod_{j=c+1}^{d} B\left(\nu_2 + \sum_{m=j}^{k} a_m + b_{m+1}, b_j \right) \Big/ \prod_{j=c+1}^{d} B\left(\sum_{m=j}^{k} a_m + b_{m+1}, b_j \right) \right] \tag{2.8}$$

for $c \leq d$ and $\nu_1, \nu_2 > 0$.

The posterior joint distribution (2.4) can thus be expressed as

$$\sum_{\ell_1=0}^{s_1} \cdots \sum_{\ell_k=0}^{s_k} \frac{W(\underset{\sim}{\ell}, \underset{\sim}{s}, \underset{\sim}{n}, \underset{\sim}{\alpha}, \beta)}{\bar{W}} \left[\prod_{j=1}^{k} \left\{ u_j^{\ell_j + n_j - s_j} (u_{j-1} - u_j)^{\beta\alpha_j - 1} \right\} u_k^{\beta\alpha_{k+1}-1} \Big/ \prod_{j=1}^{k} B\left(\sum_{m=j}^{k} (\ell_m + n_m - s_m + \beta\alpha_{m+1}), \beta\alpha_j \right) \right] \tag{2.9}$$

where

$$W(\underset{\sim}{\ell}, \underset{\sim}{s}, \underset{\sim}{n}, \underset{\sim}{\alpha}, \beta) = \prod_{j=1}^{k} \binom{s_j}{\ell_j} (-1)^{\ell_j} B\left(\sum_{m=j}^{k} (\ell_m + n_m - s_m) + \beta\alpha_{m+1}, \beta\alpha_j \right) \tag{2.10}$$

and

$$\bar{W} = \sum_{\ell_1=0}^{s_1} \cdots \sum_{\ell_k=0}^{s_k} W(\underset{\sim}{\ell}, \underset{\sim}{s}, \underset{\sim}{n}, \underset{\sim}{\alpha}, \beta) \, .$$

Once the weights $W(\underset{\sim}{\ell}, \underset{\sim}{s}, \underset{\sim}{n}, \underset{\sim}{\alpha}, \beta)$ and $\bar{W}$ are obtained, posterior joint moments for (2.9) are obtained as a weighted combination of moments of the form (2.8),

$$\sum_{\ell_1=0}^{s_1} \cdots \sum_{\ell_k=0}^{s_k} \frac{W(\underset{\sim}{\ell}, \underset{\sim}{s}, \underset{\sim}{n}, \underset{\sim}{\alpha}, \beta)}{\bar{W}} E[u_c^{\nu_1} u_d^{\nu_2} | (\underset{\sim}{\ell} + \underset{\sim}{n} - \underset{\sim}{s}), (\beta \cdot \underset{\sim}{\alpha})] \tag{2.11}$$

Though the expressions (2.9), (2.10), and (2.11

are closed form expressions, they may be difficult to evaluate. The time required to evaluate such expressions is a function of $\underline{s}$ (and thus $\underline{n}$) and k. In addition, when the n_i are large, computer evaluation of the required beta function can lead to significant numerical error. This is due to the fact that because of the summations involved, the required argument values often far exceed the maximum allowable values specified for accuracy. One alternative is to use the regeneration formula for the gamma function and factor thus greatly simplify expressions (2.9), (2.10), and (2.11). In so doing we may rewrite (2.10) and (2.8) as

$$W(\underline{\ell},\underline{s},\underline{n},\underline{\alpha},\underline{\beta}) = \prod_{j=1}^{k} \binom{s_j}{\ell_j} (-1)^{\ell_j} \prod_{f=1}^{z_j} \left[1 - \frac{\beta\alpha_j}{f - 1 + \sum_{m=j}^{k} (n_m - s_m) + \sum_{m=j}^{k+1} \beta\alpha_m} \right] \tag{2.12}$$

where $z_j = \sum_{m=j}^{k} \ell_m$ and

$$E[u_c^{\nu_1} u_d^{\nu_2} \mid \underline{a},\underline{b}] = \prod_{j=1}^{c} \prod_{f=1}^{\nu_1+\nu_2} \left[1 - \frac{b_j}{f - 1 + b_{m+1} + \sum_{m=j}^{k} a_m + b_m} \right] \prod_{j=c+1}^{d} \prod_{f=1}^{\nu_2} \left[1 - \frac{b_j}{f - 1 + b_{m+1} + \sum_{m=j}^{k} a_m + b_m} \right] \tag{2.13}$$

respectively. A numerical problem may still exist for large n_i in that the evaluation involves differences of products of numbers very close to 1. While further numerical techniques can be employed we suggest posterior approximation techniques as an alternative.

3. THE TIERNEY - KADANE POSTERIOR APPROXIMATION TECHNIQUE

All the posterior quantities of Section 2 can be obtained using the Tierney - Kadane approximation technique for evaluating

$$\frac{\int \cdots \int U(\underline{u})\, e^{\Lambda(\underline{u})}\, du_1 \cdots du_k}{\int \cdots \int e^{\Lambda(\underline{u})}\, du_1 \cdots du_k} \tag{3.1}$$

where

$\Lambda(\underline{u}) = \frac{L(\underline{u}) + \Pi(\underline{u})}{N}$

$L(\underline{u})$ = Log of the likelihood

$\Pi(\underline{u})$ = Log of the prior joint distribution for $\underline{u}$

$U(\underline{u})$ = function of $\underline{u}$.

3.1 General Results

Based on the Laplace's method, the approximation to (3.1) is given by

$$\left[\frac{\det(\sum^{*})}{\det(\sum)}\right]^{\frac{1}{2}} \exp\left\{ N \left[\Lambda^{*}(\underline{\hat{u}}^{*}) - \Lambda(\underline{\hat{u}}) \right] \right\} \tag{3.2}$$

where $\Lambda^{*} = \eta(\underline{\hat{u}}) + \Lambda$ and $\eta(\underline{\hat{u}}) = \text{Log}(U(\underline{\hat{u}}))$ and $\underline{\hat{u}}^{*}$ ($\underline{\hat{u}}$) is the joint mode of Λ^{*} (Λ) and $\sum^{*}$ ($\sum$) is minus the inverse Hessian of $\Lambda^{*}(\Lambda)$ evaluated at $\underline{\hat{u}}^{*}$ ($\underline{\hat{u}}$) and $N \sum_{j=1}^{k} n_j$. Note that due to the fact that the prior distribution is defined over the simplex

$$1 = u_0 \geq u_1 \geq u_2 \geq \cdots \geq u_k \geq u_{k+1} = 0,$$

the integration is also subject to this restriction and therefore so is selection of the joint modal values.

For obtaining the posterior marginal distribution of say u_t which was not discussed in Section 2 due to its increased complexity, we note that

$$p(u_t \mid \underline{s},\underline{n}) = \frac{\int \cdots \int e^{\Lambda(\underline{u})}\, du_1 \cdots du_{t-1}\, du_{t+1} \cdots du_m}{\int \cdots \int e^{\Lambda(\underline{u})} du_1 \cdots du_m} \tag{3.3}$$

For a given u_t let $\underline{\hat{u}}(u_t)$ be the mode of Λ for u_t fixed at its value and call this function Λ_t. Note that $\underline{\hat{u}}(u_t)$ is an m - 1 by 1 vector and let $\sum_t$ denote the *minus inverse* Hessian of Λ_t evaluated at $\underline{\hat{u}}(u_t)$ (this should have 1 less rank then that of $\sum$). Then the posterior marginal distribution of u_t is given by

$$p(u_i \mid \text{Data}) = \left\{\frac{\det(\sum_i)}{2\pi N \det(\sum)}\right\}^{\frac{1}{2}} \exp\left\{ \Lambda_i(\hat{\underline{u}}(u_i)) - \Lambda(\hat{\underline{u}}) \right\} \tag{3.4}$$

Thus all the desired posterior quantities of Section 2 may be obtained and in addition we may obtain the posterior marginal distribution for each u_i.

3.2 The Optimization Problem

Usually the major difficulty in using the Tierney Kadane method is in solving two separate optimization problems. This is particularly true when dealing with a constrained problem as we have here. However, we can show that for this problem the derivative expressions are straightforward and in addition, with a simple transformation of the parameters we can convert our problem to an unconstrained problem.

The log of the posterior distribution $N \cdot \Lambda(\underline{u})$ given by

$$(s_1 + \beta\alpha_1 - 1)\,\text{Ln}\{1 - u_1\} + \sum_{j=2}^{k} s_j\,\text{Ln}\{1 - u_j\}$$

$$+ \sum_{j=1}^{k-1} (n_j - s_j)\text{Ln}\{u_j\} + (n_k - s_k + \beta\alpha_{k+1} - 1)\text{Ln}\{u_k\}$$

$$+ \sum_{j=2}^{k} (\beta\alpha_j - 1)\text{Ln}\{u_{j-1} - u_j\} \tag{3.5}$$

Denoting (3.5) as Φ, we may find the mode of the posterior by setting $\frac{\partial \Phi}{\partial u_i} = 0$, $i = 1, \ldots, k$, and solving with

$$\frac{\partial \Phi}{\partial u_1} = \frac{(s_1 + \beta\alpha_1 - 1)}{1 - u_1} + \frac{n_1 - s_1}{u_1} + \frac{\beta\alpha_2 - 1}{u_1 - u_2}$$

$$\frac{\partial \Phi}{\partial u_i} = \frac{s_i}{1 - u_i} + \frac{n_i - s_i}{u_i} + \frac{\beta\alpha_i - 1}{u_{i-1} - u_i} + \frac{\beta\alpha_{i+1} - 1}{u_i - u_{i+1}}$$

$$i = 2, \ldots, k-1$$

$$\frac{\partial \Phi}{\partial u_k} = \frac{s_k}{1 - u_k} + \frac{n_k - s_k + \beta\alpha_{k+1} - 1}{u_k} + \frac{\beta\alpha_k - 1}{u_{k-1} - u_k} \tag{3.6}$$

The second derivatives are given by

$$\frac{\partial^2 \Phi}{\partial u_1^2} = -\frac{(s_1 + \beta\alpha_1 - 1)}{(1 - u_1)^2} - \frac{n_1 - s_1}{u_1^2} - \frac{\beta\alpha_2 - 1}{(u_1 - u_2)^2}$$

$$\frac{\partial^2 \Phi}{\partial u_i^2} = -\frac{s_i}{(1 - u_i)^2} - \frac{n_i - s_i}{u_i^2} - \frac{\beta\alpha_i - 1}{(u_{i-1} - u_i)^2} - \frac{\beta\alpha_{i+1} - 1}{(u_i - u_{i+1})^2} \qquad i = 2, \ldots, k-1$$

$$\frac{\partial \Phi}{\partial u_i \partial u_{i-1}} = -\frac{\beta\alpha_i - 1}{(u_{i-1} - u_i)^2} \qquad i = 2, \ldots, k-1$$

$$\frac{\partial \Phi}{\partial u_i \partial u_{i+1}} = -\frac{\beta\alpha_{i+1} - 1}{(u_i - u_{i+1})^2} \qquad i = 2, \ldots, k-1$$

$$\frac{\partial^2 \Phi}{\partial u_k^2} = -\frac{s_k}{(1 - u_k)^2} - \frac{n_k - s_k + \beta\alpha_{k+1} - 1}{u_k^2} - \frac{\beta\alpha_k - 1}{(u_{k-1} - u_k)^2} \tag{3.7}$$

To convert to an unconstrained optimization program we select the following reparameterization

$$u_i = \prod_{j=1}^{i} \left[1 + \frac{1}{\exp(\theta_j)}\right] \qquad i = 2, \ldots, k \tag{3.8}$$

or inversely,

$$\theta_i = \ln\left\{\frac{u_{i-1} - u_i}{u_{i-1}}\right\} \qquad i = 2, \ldots, k \tag{3.9}$$

Maximization of the reparameterized problem is facilitated by replacing (3.6) with

$$\frac{\partial \Phi}{\partial \theta_i} = \sum_{j=1}^{k} \left[\frac{\partial \Phi}{\partial u_j}\right]\left[\frac{\partial u_j}{\partial \theta_i}\right] \qquad i = 1, \ldots, k.$$

The additional terms $\frac{\partial u_j}{\partial \theta_i}$ which are easily obtained as

$$\frac{\partial u_j}{\partial \theta_i} = \begin{cases} \dfrac{u_j}{1 + \exp(\theta_i)} & i \le j \\ 0 & i > j \end{cases} \tag{3.10}$$

Under reasonable choices for the prior parameters, specifically for $\beta\alpha_i \ge 1$ for all i, the function Φ is guaranteed to be concave and this

appears to be true for any definition of the prior parameters provided the n_i are large enough. Similar arguments can be used to show that for reasonable functions $U(\underline{u})$ the function $\Psi^* = N(\Lambda^*)$ is also concave. Thus the optimization is really no problem.

4. CONCLUSIONS

The Tierney Kadane approximation technique appears to make reason iii of Section 1 considerably more true and allows for the application of the ordered Dirichlet distribution in large sample situations. In a future paper we will give the results and comparison of numerical calculation, numerical integration, and the Tierney Kadane approximation technique, for obtaining posterior quantities for various combinations of n_i, s_i and k.

REFERENCES

[1] Conor, R. and Mosimann, J. (1969). Concepts of Independence for Proportions with a Generalization of the Dirichlet Distribution. *Journal of the American Statistical Association*, Vol. 64, pp. 194 - 206.

[2] Lochner, R. (1975). A Generalized Dirichlet Distribution in Bayesian Life Testing. *Journal of the Royal Statistical Society, Series B*, Vol. 37, pp. 103 - 113.

[3] Mazzuchi, T. and Singpurwalla, N. (1982). The U. S. Army (BRL's) Kinetic Energy Penetrator Problem: Estimating the Probability of Response for a Given Stimulus. *Proceedings of the Twenty Seventh Conference on the Design of Experiments in Army Research Development and Testing*, ARO Report 82-2, pp. 27 - 58.

[4] Mazzuchi, T. and Singpurwalla, N. (1983). A Bayesian Approach for Inference for Monotone Failure Rates. *Statistics and Probability Letters*, Vol. 3, pp. 135 - 141.

[5] Mazzuchi, T. (1986). A Bayesian Approach to Inference from Accelerated Life Tests. Submitted for publication.

[6] Ramsey, F. (1972). A Bayesian Approach to Bioassay. *Biometrics*, Vol. 28, pp. 841 - 858.

[7] Tierney, L. and Kadane, J. (1986). Accurate Approximations for Posterior Moments and Marginal Densities. *Journal of the American Statistical Association*, Vol. 81, pp. 82 - 06.

COMPARISON OF "LOCAL MODEL" STATISTICAL CLASSIFICATION METHODS

Daniel Normolle, University of Michigan

Introduction

The different methods of statistical classification may be divided into two groups: those which require assumptions concerning the class-conditional distribution functions (e.g., linear discrimination, logistic regression); those which classify observations depending upon the class membership of the nearby observations, such as nearest neighbor classification and CART. This paper is concerned with a comparison of several of the latter, "local" methods, taken from a Monte Carlo study performed at the State University of New York at Binghamton and the University of Michigan which examines various aspects of 22 statistical classification methods over 12,000 data sets. The estimated rates of correct classification for the various methods, analysis of the differences in performance of the methods, and characteristics of the optimization techniques used in the various methods are presented. In particular, the use of cross-validation to select the neighborhood size in the nearest neighbor method is discussed.

Methods

The local classification methods have in common the rule that, in general, observations are assigned to the same class as their neighbors. Each local classification method has a different definition of what a neighborhood is, and these different definitions constitute the differences between the methods. Each of the methods requires that the neighborhood size be optimized by some method to prevent neighborhoods which are too small and hence overfit the data, or which are too large and miss important features of the measurement space. For each of the following methods, the neighborhood process will be described, along with the method to optimize the neighborhood size.

The dimension of the measurement space is represented by p, the number of classes by d, and the number of observations in the training sample by $n = \sum_{i=1}^{d} n_i$. A p-dimensional observation is denoted by $\mathbf{x}$, with j^{th} component x^j. The proportion of the population in the i^{th} class is π_i, and the class-conditional density function is written $f_i(\cdot)$. The sample mean of the i^{th} class is written $\bar{\mathbf{x}}_i$, and the common sample covariance matrix is $\mathbf{S}$.

The *classification tree* method (OTREE), a refinement of an older technique called classification by statistically equivalent blocks, is described in detail in Breiman et al (1984). The measurement space is recursively partitioned into rectangles by cuts perpindicular to the measurement axes, and a classification rule is assigned to each rectangle based on the class membership of the observations within the rectangle. The recursion is halted when the rectangles contain observations from only one class, or contain only one observation. The structure of the classification rule is represented by a binary tree, where the cuts are placed at the non-terminal nodes, and each rectangle is assigned to a terminal node.

The recursive rule as described tends to construct trees which overfit the data, resulting in over-optimistic estimates of classification rates on training data, and poor performance on subsequent data sets. The tree size is optimized by growing ten auxiliary trees, each based on nine-tenths of the original training set, and then using the hold-out sample consisting of the remaining data points to estimate the true correct classification rate. The trees are used to determine the value of a *cost-complexity* parameter which penalizes both overcomplicated trees and high misclassification rates. The value of this parameter is used to "trim" the main tree, by combining the rectangles with small numbers of observations to achieve a tree which represents the data, but does not overfit it.

The *kernel PDF estimation* method of statistical classification (KERNEL) directly estimates $f_i(\mathbf{x})$ using only weak assumptions about the functional form of the $f_i(\cdot)$, and then assigns class-membership depending upon the values of the estimated density functions. The application of density estimation to statistical classification is described in Hand (1981). The density estimator at $\mathbf{x}=(x^1,\ldots,x^p)'$ is:

$$\hat{f}_i(\mathbf{x}) = \frac{1}{n_i \prod_{j=1}^{p} h_{ij}} \sum_{k=1}^{n_i} \left(\prod_{j=1}^{p} K\left(\frac{x^j - x^j_{ik}}{h_{ij}} \right) \right)$$

where $K(\cdot)$ is a symmetric, univariate probability density function and x^j_{ik} is the j^{th} element of the k^{th} member of the i^{th} class in the training sample. The use of a density function for K ensures that nearby points will contribute more to the density

estimate than distant points. The kernel function of Epanechnikov (1969),

$$K(u)=\max\{\frac{3}{4\sqrt{5}} - \frac{3u^2}{20\sqrt{5}},0\},$$

which asymptotically minimizes the mean integrated square error for a large class of univariate density functions (Tapia and Thompson, 1978) is used; it has the additional advantage of a bounded support, which reduces computational cost.

The smoothing parameters h_{ij} are calculated independently for each class i and coordinate j by an iterative method. The process is initialized by the range of the training sample:

$$h_{ij}^0 = \max\{x_{i1}^j,\ldots,x_{in_i}^j\} - \min\{x_{i1}^j,\ldots,x_{in_i}^j\},$$

and is updated according to:

$$h_{ij}^{s+1} = n_i^{-1/5}\alpha(K)\beta(h_{ij}^s),$$

where:

$$\alpha(K) = (\frac{12\sqrt{5}}{100})^{1/5}$$

and, if $\hat{f}_{ij}^s(\cdot)$ is the estimate of $f_{ij}^s(\cdot)$ using the training sample and h_{ij}^s,

$$\beta(h_{ij}^s) = (n\int[\hat{f}_{ij}^{s\prime\prime}(y)]^2dy)^{-1/5}$$

The resulting density estimates for each class are then calculated at a test point $\mathbf{x}$, and the test point is assigned to the class having the largest estimate.

The *k-nearest neighbor* method (KNN, Fix and Hodges, 1951) is supplied by the analyst with an integer k, and then classifies a point $\mathbf{x}$ according to the class memberships of the k observations from the training sample which are closest in the measurement space to $\mathbf{x}$. Distance is measured by the Mahalanobis distance,

$$d(\mathbf{x},\mathbf{y})=(\mathbf{x}-\mathbf{y})'\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y}).$$

The choice of the size of the neighborhood, determined by k, is as problematic with the k-nearest neighbor method as with those previously mentioned. Here, cross-validation is used to determine k, producing a *cross-validated k-nearest neighbor* (XKNN). Each observation $\mathbf{x}_{im}$ in the training set is classified using the other points in the training set as follows. The Mahalanobis distance to every other point $\mathbf{x}_{jl}$ in the training set is first calculated:

$$d_{jl} = (\mathbf{x}_{im}-\mathbf{x}_{jl})'(\mathbf{S}_{im})^{-1}(\mathbf{x}_{im}-\mathbf{x}_{jl}),$$

$$(j=1,\ldots,d \text{ and } l=1,\ldots,n_j)$$

(where $(\mathbf{S}_{im})^{-1}$ is the inverse of the common covariance matrix with the $im^{\underline{th}}$ observation removed. This vector of n-1 distances is sorted along with the class memberships of the training observations. Then, starting with k=1, a running tally is kept of the class memberships. For each k, the k class memberships vote, resulting in an class estimate for $\mathbf{x}_{im}$ for each value of $1\leq k\leq n-1$. This procedure is repeated for all the points in the training sample, and the number of points correctly classified for each value of k is accumulated. k^* is then selected as that particular value of k which maximizes the number of correctly classified training points (actually, a three-point moving average is maximized), yielding an optimized neighborhood size for the nearest-neighbor method. Each observation in the testing sample is then classified using the k^* nearest neighbors in the training sample.

The last method is not included as a "local model" method, but because it represents an interesting bridge to the global methods in the original study. Each observation in the training sample is replaced, coordinate-by-coordinate, by its *normal score*. The normal score of the $i^{\underline{th}}$ largest value in a set of numbers $\{x_1,\ldots,x_n\}$ equals $\Phi^{-1}\left(\frac{i}{n+1}\right)$, where $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function. The testing observations are ordered independently of the training observations. The *conditional* discriminant function performs a linear or quadratic discriminant analysis conditional on a test of the hypothesis

$$H_0: \Sigma_1 = \Sigma_2,$$

where Σ_i is the dispersion matrix of the $i^{\underline{th}}$ class. $\mathbf{S}_i$ is the within-class sample covariance matrix, calculated from the transformed data, and $|\mathbf{S}|$ is the determinant of $\mathbf{S}$. The test used in the simulations is a special case (for $n_1=n_2$ and $d=2$)

of Box's (1949) modification to Wilks' Λ: calculate

$$\rho = 1 - \frac{(2p^2 + 3p - 1)}{2(n-2)(p+1)},$$

$$\Lambda = \frac{|S_1||S_2|}{|S|},$$

and reject the hypothesis of equality of dispersion matrices if $-2 \cdot \text{rho} \cdot \log(\Lambda)$ exceeds the 90^{th} percentile of a χ^2 distribution with $p(p+1)/2$ degrees of freedom. The symbol NCDF is used to represent the use of the conditional discriminant function on the normal scores of the data.

Since Bayes rule maximizes the expected probability of correct classification (Glick, 1971), it shall be used as a benchmark against which the other methods will be compared. The version of Bayes rule used here, which is less than general but sufficient for the circumstances of the experiment, where $d = 2$ and $\pi_1 = \pi_2 = \frac{1}{2}$, is:

$$\text{Classify } \mathbf{x} \in i \text{ if } f_i(\mathbf{x}) = \max\{f_1(\mathbf{x}), f_2(\mathbf{x})\},$$

where $f_1(\cdot)$, $f_2(\cdot)$ are known class-conditional probability density functions. Bayes rate, the proportion of observations correctly classified using Bayes rule, will be estimated by calculating the class-conditional density functions at each testing sample point, which is possible since these density functions are known precisely in a simulation experiment. The correct classification rates of the individual methods will be reported as the percentage of the Bayes rate.

The full details of the Monte Carlo Experiment are presented in Normolle (1987). The simulations are written in FORTRAN, compiled on the IBM H optimizing compiler, and executed on an IBM 4381 at the State University of New York at Binghamton, and on the IBM 3090-400 at the University of Michigan.

The experiment is a fully-crossed $5 \times 2 \times 2 \times 3 \times 2$ design with five levels of variation: distribution type (Gaussian, Cauchy, lognormal, bimodal, uniform); dimension (2, 6); training sample size (40, 80, 160); between-class separation (low, high); within-class dispersion (equal, unequal). A multiplicative congruential generator with multiplier 7^5 and base $2^{31} - 1$ generates the primary [0,1] random variates, which are then transformed to specific standardized distributions by well-known methods (e.g., Gaussian variates are obtained by the Box-Muller transformation) described in the thesis. Multiplication by rotation matrices and translation by location vectors determine specified population location and dispersion, chosen to achieve (within 1%) a predetermined Bayes' rate in the population. Each design point was replicated 100 times, for a total of 12,000 training sets. All classification methods are calibrated on every training set, and evaluated by their correct classification rate on a test set of 1000 observations associated with the design cell.

Results

The result of optimization on the classification tree and nearest neighbor methods is tested by comparing the optimized to the non-optimized version. The non-optimized classification tree (TREE) is the tree grown on the training sample without cross-validation pruning. The XKNN is compared to the KNN rule, where k is the smallest odd integer greater than the square root of the training sample size.

It is seen from Table 1 that pruning has a larger effect on the higher-dimensional data, and that the effect increases with the training sample size on both the 2-and 6-dimensional sets. Analysis of variance on the difference between TREE and OTREE (not displayed) produces significant main effects for all the experimental variables.

Cross-validation selects neighborhoods which are larger than the square root rule (Table 2) and

Table 1.
Mean Percent of Bayes Rate

p	n	%of Bayes Rate		Paired Comp. t
		TREE	OTREE	
2	40	86.6	86.5	−2.0
	80	89.7	90.1	5.4**
	160	92.4	93.6	16.6**
6	40	82.6	83.1	8.0**
	80	87.6	89.4	18.8**
	160	91.9	93.8	21.9**

** $p < 0.0001$

Table 2.
Mean Values of Cross-Validated k

p	n	Mean k
2	40	9.28
	80	17.18
	160	30.32
6	40	9.27
	80	17.86
	160	31.51

Table 3.
Mean Values of k

Distribution	Mean k
Normal	21.60
Cauchy	12.25
Lognormal	14.02
Bimodal	24.05
Uniform	24.25

seem to be only slightly affected by the dimension of the data. The neighborhoods are smaller on the heavier-tailed distributions (Cauchy and Lognormal, Table 3), and larger on the shorter-tailed (Uniform and Bimodal). An analysis of variance (not shown) demonstrates that all the design variables except the equality of the within-class dispersion matrices significantly effect the difference in the classification rates between the KNN and XKNN rules. As seen in Table 4, cross-validation degrades the perfomance of the nearest-neighbor method on the sparsest data (n=40 and 80, p=6), but as the concentration of data increases, the improvement in classification increases substantially. The increase is pronounced in situations where the classes are not well-separated (Table 5), while the cross-validated and square root values of k produce essentially the same results when the classes are already well-separated.

The iterative process to calculate smoothing parameters for KERNEL tends to produce values which are smaller than optimal (the optimal values can be determined exactly for some of the known class-conditional densities). A cross-validation method of smoothing parameter estimation is currently being implemented to remedy this situation.

Tables 6 and 7 present the mean and

Table 4.
Mean Percent of Bayes Rate

p	n	%of Bayes Rate		Paired Comp. t
		KNN	XKNN	
2	40	91.8	92.8	10.7 **
	80	93.4	94.8	16.8 **
	160	94.6	96.5	21.8 **
6	40	90.2	88.2	−14.4 **
	80	91.8	91.5	−3.1 *
	160	91.9	93.8	13.8 **

* $p<0.01$
** $p<0.0001$

minimum percentage of Bayes rate obtained by the four methods considered, ordered by dimension and sample size. Each mean and minimum displayed are based on 2000 observations. For the 2-dimensional data, XKNN displays the highest average standardized classification rate of all 22 methods. As the number of measurement variables increases, the efficacy of NCDF, as measured by the mean, actually increases, while the KERNEL and XKNN methods are degraded. At the lowest sample size, OTREE's performance is inferior to the other methods, but it improves at n=80 and n=160. In addition, OTREE's response rate at n=160 is relatively unchanged between p=2 and p=6, unlike the other local methods, which seem to degrade quickly as the number of measurement variables increases.

Table 5.
Mean Percent of Bayes Rate

Separation	%of Bayes Rate		Paired Comp. t
	KNN	XKNN	
Low	89.3	90.7	19.4 **
Hi	95.3	95.2	1.3

** $p<0.0001$

Table 6.
Mean Percent of Bayes Rate

p	n	%of Bayes Rate			
		OTREE	KERNEL	XKNN	NCDF
2	40	86.5	92.1	92.8[a]	89.4
	80	90.1	93.7	94.8[a]	91.8
	160	93.6	95.1	96.5[a]	91.8
6	40	83.1	87.1	88.2	91.1[a]
	80	89.4	88.8	91.5	93.4[a]
	160	93.8	90.7	93.8	94.8

[a] Best of All Methods

Table 7.
Minimum Percent of Bayes Rate

p	n	Minimum of Bayes Rate			
		OTREE	KERNEL	XKNN	NCDF
2	40	49.9	70.4	71.0[a]	40.6
	80	63.7	70.3	73.0[a]	43.9
	160	74.6	78.0	79.6[a]	35.9
6	40	34.6	52.8	17.4[b]	42.2
	80	50.6	56.4	51.2	41.6
	160	69.2[a]	58.6	55.2	44.2

[a] Best of All Methods
[b] Worst of All Methods

XKNN performs well compared to all other methods with respect to the minimum correct classification rate at $p=2$, and competes reasonably at $p=6$, except for a disaster which occurred at $n=40$. KERNEL is notable in that the minimum rate does not decrease as far at small sample sizes at $p=2$ and $p=6$ as NCDF and OTREE. KERNEL is the minimax method over all 12,000 observations.

Table 8 displays the mean percentage of Bayes rate obtained by distribution type of the sample data. Each mean is based on 2400 observations. NCDF, which works remarakbly well with elliptical distributions (Normal and Cauchy), even if they are heavy-tailed, breaks down substantially when presented with data from the very skewed Lognormal distribution.

Analyses of variance (not shown) produce very significant main effects ($p<0.0001$) on the classification rates of the four methods studied. The design variables account for 42% to 52% of the variance of the local model methods, and 72% of NCDF.

Table 8.
Mean Percent of Bayes Rate

Dist. Type	%of Bayes Rate			
	OTREE	KERNEL	XKNN	NCDF
Normal	90.8	93.6	97.1	100.0
Cauchy	98.3	96.6	97.3	103.3
Lognormal	82.8	81.0	82.3	63.4
Bimodal	92.2	96.5	97.2	99.5
Uniform	82.9	88.6	90.8	93.9[a]

[a] Best of All Methods

Discussion and Conclusions

Generalizations from a Monte Carlo experiment are, of course, problematic, so we proffer the following conclusions and recommendations with the usual caveats.

While, at the sample sizes considered, cross-validation is of some benefit, the computational cost is high and the value in classification power limited. Thus, cross-validation requires training samples at least as large as the biggest considered here to be effective even on very (e.g., $p=2$ or 3) low-dimensional data.

Since it is based on the marginal empirical distribution functions, OTREE performs better on the higher-dimensional data than does either the XKNN or KERNEL. However, the low mean and minimum rates at $n=40$ and $n=80$ suggest that OTREE is not appropriate at these small sample sizes, but that once an adequate sample size (say, 150 training observations) is obtained, more variables may be included in the analysis than with competing local model methods.

KERNEL and XKNN offer higher average performance at the lower dimension. KERNEL is the minimax classifier over the entire experiment.

NCDF shows promise on the sparse, higher-dimensional data when the sample size is too small for the effective performance of the classification tree, but is subject to degraded performance when the data are very skewed.

As a group, the local model methods are strong at $p=2$; if the analyst is unable to make any assumptions about the data, sample sizes like those considered in this study will yield good results, especially with the cross-validated nearest neighbor. The classification tree method requires larger sample sizes than the other methods even with two dimensions, but can tolerate more variables once this barrier is overcome. The cost of all of these "assumption-free" methods increases rapidly with the number of dimensions, so that either some dimension reduction technique must be applied to the data before a local-model method is applied, the sample size must be quite large, or a rank-based or robust global alternative must be employed.

References

Box, G. (1949), "A General Distribution Theory for a Class of Likelihood Criteria", *Biometrika*, 40, 317–346.

Breiman, L., Friedman, J., Olsen, R., and Stone, C. (1984), *Classification and Regression Trees*, Belmont, California: Wadsworth International Group.

Epanechnikov, V. (1969), "Non-parametric Estimation of a Multivariate Probability Density", *Theory of Probability and Its Applications*, 14, 153–158.

Fix, E. and Hodges, J., "Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties", Technical Report, Randolph Field, Texas: USAF School of Aviation Medicine.

Hand, D. (1982), *Kernel Discriminant Analysis*, London: Research Studies Press.

Glick, N. (1972), "Sample-Based Classification Procedures Derived from Density Estimators", *Journal of The American Statistical Association*, 67, 116–122.

Normolle, D. (1988), *Comparing the Performance of Classification Methods*, PhD Thesis, Binghamton: State University of New York.

Tapia, R., and Thompson, J. (1978), *Nonparametric Density Estimation*, Baltimore: The Johns Hopkins University Press.

An Example of the Use of A Bayesian Interpretation of MDA Results

James R. Nolan, Siena College

This article is concerned with the interpretation of multiple discriminant analysis results. Specifically, a demonstration will be made of the usefulness of Bayesian methods for enhancing the utility of the multiple discriminant results.

The primary objective of discriminant analysis is to classify cases into two or more groups. An implicit assumption for using this technique is that the groups can be differentiated based on a combination of multivariate normal variables. In addition, if the variances and covariances of the independent variables are equal, or nearly so, a linear classification model is optimal.

Thus,

$$D_j = b_1 x_{1j} + b_2 x_{2j} + \ldots + b_n x_{nj}$$

where j = group number
b_i = coefficient

The general procedure for conducting an analysis using the discriminant method is as follows:

(1) a priori definition of a sample of cases in each group.
(2) definition of the variables which are thought to account for intergroup differences.
(3) submission of the data to an MDA (multiple discriminant analysis) algorithm.
(4) determination of the "cutting point" or critical discriminant score which will separate the groups.
(5) ultimately, you want the probability of specific group membership.

Once the procedure is complete, there are various methods used to determine the value of the resulting equation - value as far as statistical significance is concerned as well as its inference or predictive ability. Some of these procedures are:

eigenvalue = btwn. group SS/within group SS

The larger the eigenvalue, the better the equation is able to differentiate.

canonical correlation = the degree of association between the discriminant scores and the groups.

The higher this value, the better.

confusion matrix = percentage of cases correctly and incorrectly classified.

The major problem with this last method is that you are using the very same cases used for constructing the equation to determine the value of the equation - a very biased view results. One way to get around this is to divide the data into two groups at the beginning of the analysis; then use one group to construct the equation and the other group to determine its value. The problem here is that you lose one half of your original data when constructing the equation. A better procedure is to employ the "jacknife" method (alternately referred to as the "leaving one out" method) whereby you construct the equation using all but one of your cases and then proceed to classify that "left out" case. After doing this for all cases, you have a much less biased view of the value of the discriminant equation.

The major problem with stopping the analysis at this point is that two important items are being ignored:

(1) prior probabilities of group membership.
(2) incorporating additional information about cases.

These two items can be included in the analysis if we utilize Bayes' Rule. If the "cutting point" or critical discriminant score is placed midway between the mean discriminant D scores for each group (in the two group case), the implied objective is to equalize the probabilities of misclassifying the cases.

In many situations, we know that there is a higher probability a case belongs to one group versus the other. If your objective is to minimize misclassification, period, then the cutting point should be moved toward the mean D score of the smaller group.

How far should the cutting point be moved? One could alternately try many different cutting points to find the best. Needless to say, this would be very time consuming. Bayes' rule will come in handy here, but we still need some more information; suffice it to say that we should be aware of prior probabilities P(group i).

We are still not at the point where we can determine the optimal solution (equation). Consideration of additional information available for each case will help us. To take advantage of the additional information available, we need to assess the likelihood of the additional information under different circumstances.

For example, if the discriminant function scores are normally distributed for each of two groups, and the parameters of the distribution can be estimated, it is possible to calculate the probability of obtaining a different discriminant function value if the case is a member of group one or group two.

This probability is called the conditional probability of the discriminant score (D), given the group, P(D/G(i)). To calculate the probability, the case is assumed to belong to a particular group, and the probability of an observed score given membership in the group is estimated using the normal distribution.

Finally, this information about group membership and the conditional probability of obtaining a discriminant score given a certain group membership, can now be combined using Bayes' rule; this will help us to determine what we were interested in all along - namely, how

likely membership in the various groups is, given the available information - referred to as the posterior probability.

To demonstrate the usefulness of this procedure, we can look at the following example: we wish to determine the financial measures that are useful for predicting the financial health of a hospital. The research design is now detailed:

(1) Financial data was collected on 48 New York State hospitals over a three year period 1980-82.
(2) These hospitals were selected based upon a New York State management group's opinion as to the most fiscally sound and the most fiscally distressed hospitals in the state. Their decisions did not result from consideration of the proposed explanatory financial variables.
(3) The three year average (mean) and three year variation (standard deviation) was calculated for each of the 72 possible explanatory variables. The purpose of these calculations is to obtain measures that indicate trends early enough to do something about them, i.e. hospital management has the opportunity to make changes to improve the fiscal health of the institution.
(4) The most statistically significant explanatory variables were identified and the discriminant scores were calculated for each hospital; the "jacknife" procedure was used to determine the classification of the sample hospitals.

At this stage, we have an equation that can be used for predicting the financial classification of a hospital - fiscally stable or fiscally distressed.

Additional utility can be obtained from this equation by considering the approximate probability of group membership in the population. In this New York State hospital example, 70% of the hospitals in the state are fiscally distressed and 30% are fiscally sound. Based upon these probabilities (group membership in the population) and the additional information obtained from the calculated discriminant score for each case and its known group, the probability of each case belonging to a particular group given its calculated discriminant score, P(G(i)/D), is obtained. It is this probability that adds to, and supplements, the normal discriminant analysis results.

In summary, this additional information will help a hospital administrator determine not only whether they are in a fiscally unsound condition, but also the severity of that condition.

UNBIASED ESTIMATES OF MULTIVARIATE GENERAL MOMENT FUNCTIONS OF THE POPULATION AND APPLICATION TO SAMPLING WITHOUT REPLACEMENT FROM A FINITE POPULATION

Nabih N. Mikhail, Liberty University

Abstract

Unbiased estimates of the multivariate general moment functions of the population are obtained when sampling from finite populations. Partitions and power sums are featured. Unbiased estimates of multivariate cumulants and moment functions are obtained as examples of application.

1. Introduction

The general moment function of the finite population (gmfp) can be written in terms of power sums associated with the partitions involved. We keep the coefficients of such power sums quite general so that the results are applicable to a variety of functions.

We treat the multivariate gmfp in this paper. Univariate results are obtained by means of coalescing. The paper follows certain ideas and results given in Dwyer, Mikhail and Tracy (1978).

By giving more specific values to these coefficients, we can obtain results for multivariate cumulants (Mikhail and Malik, 1978), for multivariate moment functions (Dwyer, 1937, p.40; 1938, p.42; Mikhail and Malik, 1978) etc. and their unbiased estimates.

The purpose here is limited to the derivations of the functional forms of the gmfp of various weights, and their unbiased estimates through weight 4 are obtained. The special cases for cumulants and moment functions are obtained as applications of the theory.

Moment functions of finite population have in common the property that they may be expressed in terms of power sums (Dwyer,1938, p.104).

Power sums in this paper are denoted by () for the sample, and by $()_N$ for the finite population of size N.

The combinatorial coefficient (Dwyer and Tracy, 1964, p.1174)

associated with a partition $P = p_1^{\pi_1} \dots p_s^{\pi_s}$ (p_i distinct) of unipartite number $p = \Sigma\, p_i \pi_i$ is

$$\phi(P) = \frac{p!}{(p_1!)^{\pi_1} \dots (p_s!)^{\pi_s}\, \pi_1! \dots \pi_s!} \qquad (1.1)$$

For the multipartite number pq..., (Tracy and Dwyer, 1973, p.4), if partition $W = \left\{p_1 q_1 \dots\right\}^{\pi_1} \dots \left\{p_s q_s \dots\right\}^{\pi_s}$ (where $\left\{\right\}^{\pi_i}$ represents a part of W repeated π_i times) and $p = \Sigma\, p_i \pi_i$, $q = \Sigma\, q_i \pi_i, \dots$,

$$\phi(W) = \frac{p!q! \dots}{\left\{p_1!q_1! \dots\right\}^{\pi_1} \left\{p_2!q_2! \dots\right\}^{\pi_2} \dots \pi_1!\pi_2! \dots} \qquad (1.1')$$

The gmfp in terms of power sums is $F_{111\dots1} = \sum_U B_U (U)_N$, (1.2)

where there are r multivariate units, U is any partition of 111...1 and B_U is the coefficient of $(U)_N$.

Then, if W is any (multivariate) partition obtained by partial coalescing the columns of U, (1.2) becomes the multivariate formula

$$F_{r_1 r_2 \dots} = \sum \phi(W)\, B_W (W)_N \qquad (1.2')$$

where $r_1 + r_2 + r_3 + \dots = r$, $\phi(W)$ is the multivariate combinatorial coefficient (1.1') and B is the coefficient of $(W)_N$

The total coalescing of (1.2) and (1.2') leads to

$$F_r = \sum_P \phi(P)\, b_P (P)_N \qquad (1.2'')$$

where P is any partition of r.

We are interested in deriving the unbiased estimates E_N^* of the gmfp when sampling without replacement from a finite population of size N. Carver functions C_p are given, using different notation by Carver (1930, p.106) and Dwyer, Mikhail and Tracy (1978, p. 14,15).

$$C_p = \sum_P (-1)^{s-1} (s-1)!\, \phi(P)\, e_s \qquad (1.3)$$

where

$$e_s = \frac{n^{(s)}}{N^{(s)}}, \quad n^{(s)} = n(n-1) \dots (n-s+1). \qquad (1.4)$$

Another set of functions related to Carver functions C_p^*, given by Dwyer, Mikhail and Tracy (1978, p.16), Dwyer and Tracy (1980, p.435), and Mikhail et al (1985, p.2) follows:

$$C_p^* = \sum_P (-1)^{s-1} (s-1)!\, \phi(P)\, e_s^* \qquad (1.3')$$

where

$$e_s^* = \frac{N^{(s)}}{n^{(s)}} \qquad (1.4')$$

It is worthwhile to mention that the formulae (1.3') and (1.4') are designed for use in unbiased estimation problems.

2. The Analysis of D_U^*-Functions with more than one Subscript

Here we need the general expression $E_N^* (F_{111\dots1}) = \sum_U D_U^*(U)$ (Dwyer, Mikhail and Tracy, 1978, p.16; Mikhail and Malik, 1978, p.72; Dwyer and Tracy, 1980, p.435) where U is any partition of 111...1 and the parts of U are represented by the rows. This defines, at least implicitly, D_U as the coefficient of (U) in the E_N^* value. All special cases of the same weight (isobaric) are then obtained by coalescing. All the U under consideration are those which are partitions of 111...1 and hence have only one "1" in any column.

Deviates should not be used for the general formula. The deviate formula can then be obtained by eliminating all partitions which have one or more rows with a single "1" element. For example,

$$E_N^*(F_{11}) = E_N^* \left(B_{11}(11)_N + B_{10}\, {}_{01}\binom{10}{01}_N\right)$$

$$= (B_{11}C_1^* + B_{10}C_2^*)(11) + B_{10}C_1^*\, {}_{01}\binom{10}{01}_1$$

$$= D_{11}^*(11) + D_{10}^*\, {}_{01}\binom{10}{01}$$

since $E_N^* \left(B_{10}\, {}_{01}\binom{10}{01}_N\right) = B_{10}\, {}_{01}C_2^*(11) + B_{10}\, {}_{01}C_1^*\, {}_{1}\binom{10}{01}$

Similarly,

$$E_N^* (F_{111}) = E_N^* \left\{B_{111}(111)_N + B_{110}\, {}_{001}\binom{110}{001}_N + B_{101}\, {}_{010}\binom{101}{010}_N \right.$$

$$\left. + B_{011}\, {}_{100}\binom{011}{100}_N + B_{100}\, {}_{\substack{010\\001}}\begin{pmatrix}100\\010\\001\end{pmatrix}_N \right\}$$

$$= \left(B_{111}C_1^* + B_{110}C_2^* + B_{101}C_2^* + B_{011}C_2^* + B_{100}C_3^* \right)^{(111)}$$

$$+ \left(B_{110}C_1^* + B_{100}C_2^* \right)^{(110)(001)}$$

$$+ \left(B_{101}C_1^* + B_{100}C_2^* \right)^{(101)(010)}$$

$$+ \left(B_{011}C_1^* + B_{100}C_2^* \right)^{(011)(100)}$$

$$+ \left(B_{100}C_1^* \right)^{(100)(010)(001)}$$

$$= D^*_{111}(111) + D^*_{110}(^{110}_{001}) + D^*_{101}(^{101}_{010}) + D^*_{011}(^{011}_{100}) + D^*_{100}\,{}^{100}_{010}\,{}^{001}_{001}$$

In general this process leads to

$$E^*_H(F_{111\ldots1}) = \sum_U D^*_U(U)$$

What is needed, in general, is the specific functions in terms of B's and C's for different D_U. The evaluation of $D^*_{111\ldots1}$ is simple enough since the coefficient of the one rowed (111...1) term of the E^*_H value of $\sum_U B_U(U)_H$ is $\sum B_U C^*_u$ where U is any partition of 111...1 and u is the number of rows of U. Thus,

$$D^*_{111} = B_{111}C_1^* + B_{110}C_2^* + B_{101}C_2^* + B_{011}C_2^* + B_{100}C_3^*$$

and all the partitions of 111 with their appropriate B's and C's are included.

The evaluation of D^*_U where U consists of two or more rows is somewhat more complex. We note first that columns can be interchanged in the problem and in the result, so we may take the units in the first row in the first columns followed by the units in the second row in the second columns, etc. Thus, if we know the value of D^*_{110} and the coefficient of $^{110}_{001}$, we can interchange column 3 with column 1 and get D^*_{011} and the coefficient of $^{011}_{100}$. Also, we can get D^*_{1101}, D^*_{1011} and D^*_{0111} from D^*_{1110}; D^*_{1010} and D^*_{1001} from D^*_{1100}, etc. The other D's obtained from D^*_{11}, D^*_{111}, D^*_{1111}, etc are D^*_{110}, D^*_{1110}, D^*_{1100}, D^*_{11110}, D^*_{11100}, D^*_{11000}, etc. by column interchange.

3. Unbiased Estimates of the Multivariate gnfp

In this section we obtain unbiased estimates of the multivariate gnfp when sampling without replacement from a finite population, as $E^*_H F_{111\ldots1}$. Using the results of $^*D_U$-functions in section 2, we can derive unbiased estimated value of any population moment function when sampling without replacement from a finite population.

The formulae become quite compact if we use the integer notation suggested by Professor Dwyer (personal communications) for partitions. In this notation, the product of power sums (110) (101) (011) is written as (12, 13, 23), where the numbers 1, 2, 3, indicate the rows of $\begin{pmatrix}110\\101\\011\end{pmatrix}$ and commas separate the columns.

$$E^*_H(F_1) = D^*_1(1)$$

$$E^*_H(F_{11}) = D^*_{1,1}(1,1) + D^*_{1,2}(1,2)$$

Where the number of rows in the right is reduced to 2 with the replacement of 1 by the one in the second column by the number of the rows, we continue with the replacement of each 1 by the number of row. Hence, we get

$$E^*_H(F_{111}) = D^*_{111}(111) + D^*_{112}(112) + D^*_{121}(121) + D^*_{112}(112) + D^*_{123}(123).$$

The sum of the three middle terms can be written as

$$\sum^3 D^*_{112}(112) \quad (\text{or } \sum^3 D^*_{121}(121) \text{ or } \sum^3 D^*_{211}(211)),$$

where the Σ applies to all the different partitions resulting from interchanging columns. Similarly,

$$E^*_H(F_{1111}) = D^*_{1111}(1111) + \sum^4 D^*_{1112}(1112) + \sum^3 D^*_{1122}(1122) + \sum^6 D^*_{1123}(1123) + D^*_{1234}(1234)$$

where $\sum^4 D^*_{1112}(1112)$ indicate the four different terms resulting from the unipartition of 1112 and the interchange of column 4 with 3, 2, 1, respectively; where $\sum^3 D^*_{1122}(1122)$ indicate the three different terms resulting from the partition (1122) and the two different interchanges, etc.

The partitions featured and the number of terms in each summation are available with complete coalescing of the partitions. This is illustrated by the multinomial theorem in partition notation with the partitions written in columns:

$$(1)^4 = (4) + 4\binom{3}{1} + 3\binom{2}{2} + 6\binom{2}{1}_1 + \begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Here the multipartition reveals the numbers of unit terms in the respective rows, and the coefficient shows the number of different U which coalesce to the same partition. That number is also the combinatorial coefficient, (1.1).

4. Expressions for D^*_U

For order 2

$$D^*_{11} = B_{11}C_1^* + B_{12}C_2^*$$

$$D^*_{12} = \bar{1}_2 B^*_{11} C$$

For order 3

$$D^*_{111} = B_{1}{}_{1}{}_{1}C^*_1 + \sum^3 B_{112}C^*_2 B_{123}C^*_3$$

$$D^*_{112} = B_{112}C^*_{11} + B_{123}C^*_{21}$$

$$D^*_{123} = B_{123}C_{111}$$

For order 4

$$D^*_{1111} = B_{1111}C^*_1 + \sum^{4} B_{1112}C^*_2 + \sum^{3} B_{1122}C^*_2 + \sum^{6} B_{1123}C^*_3 + B_{1234}C^*_4$$

$$D^*_{1112} = B_{1112}C^*_{11} + B_{1123}C^*_{21} + B_{1213}C^*_{21} + B_{2113}C^*_{21} + B_{1234}C^*_{31}$$

$$D^*_{1122} = B_{1122}C^*_{11} + B_{1123}C^*_{21} + B_{2311}C^*_{21} + B_{1234}C^*_{22}$$

$$D^*_{1123} = B_{1123}C^*_{111} + B_{1234}C^*_{211}$$

$$D^*_{1234} = B_{1234}C^*_{1111}$$

For order 5

$$D^*_{11111} = B_{11111}C^*_1 + \sum^{5} B_{11112}C^*_2 + \sum^{10} B_{11122}C^*_2 + \sum^{10} B_{11123}C^*_3 + \sum^{15} B_{11223}C^*_3 + \sum^{10} B_{11234}C^*_4 + B_{12345}C^*_5$$

$$D^*_{11112} = B_{11112}C^*_{11} + B_{11123}C^*_{21} + B_{11213}C^*_{21} + B_{12113}C^*_{21} + B_{21113}C^*_{21} + B_{11223}C^*_{21} + B_{12123}C^*_{21} + B_{21123}C^*_{21} + B_{11232}C^*_{21} + B_{11234}C^*_{31} + B_{12134}C^*_{31} + B_{21314}C^*_{31} + B_{23114}C^*_{31} + B_{12314}C^*_{31} + B_{12345}C^*_{41}$$

$$D^*_{11122} = B_{11122}C^*_{11} + B_{11322}C^*_{21} + B_{13122}C^*_{21} + B_{31122}C^*_{21} + B_{11234}C^*_{22} + B_{12134}C^*_{22} + B_{21134}C^*_{22} + B_{13422}C^*_{31} + B_{12345}C^*_{\substack{3\\2}}$$

$$D^*_{11123} = B_{11123}C^*_{111} + B_{11234}C^*_{211} + B_{12134}C^*_{211} + B_{21134}C^*_{21} + B_{12345}C^*_{311}$$

$$D^*_{11223} = B_{11223}C^*_{111} + B_{11234}C^*_{211} + B_{23114}C^*_{211} + B_{12345}C^*_{221}$$

$$D^*_{11234} = B_{11234}C^*_{1111} + B_{12345}C^*_{2111}$$

$$D^*_{12345} = B_{12345}C^*_{11111}$$

etc.

5. D^*_U for Finite Version of Moment Functions $\mu_{111\ldots1}$

As an application of unbiased estimates of gmfp, the unbiased estimates of the finite version of the moment functions $\mu_{111\ldots1}$ to sampling without replacement for a finite population (Mikhail and Malik, 1978) is considered in this section.

Here the coefficients B_U for $U = 111\ldots1$ is

$$= \frac{1}{N} \quad \text{if } U = 111\ldots1 \text{ in one row}$$

$$= \frac{(-1)^{s-1}}{N^s} \quad \text{if } U = \begin{matrix} 111\ldots10000 \\ 000\ldots01000 \\ \ldots\ldots\ldots\ldots \\ 000\ldots00001 \end{matrix} \Bigg\}\, s \text{ with } 111\ldots1 \text{ in the first row and only one "1" in each row of the } (s-1) \text{ rows}$$

$$= \frac{(-1)^{s-1}(s-1)}{N^s} \quad \text{if } U = \begin{matrix} 10\ldots0 \\ 010\ldots0 \\ \ldots\ldots \\ 00\ldots01 \end{matrix} \Bigg\}\, s \text{ rows}$$

$$= 0 \text{ otherwise}$$

Then for the partition U and u in C^*_u written as a column, we have

for order 1

$$D^*_1 = \frac{1}{N} C^*_1$$

for order 2

$$D^*_{11} = \frac{1}{N} C^*_1 - \frac{1}{N^2} C^*_2$$

$$D^*_{12} = -\frac{1}{N^2} C^*_{\substack{1\\1}}$$

for order 3

$$D^*_{111} = \frac{1}{N} C^*_1 - \frac{3}{N^2} C^*_2 + \frac{2}{N^3} C^*_3$$

$$D^*_{112} = -\frac{1}{N^2} C^*_{\substack{1\\1}} + \frac{2}{N^3} C^*_{\substack{2\\1}}$$

$$D^*_{123} = \frac{2}{N^3} C^*_{\substack{1\\1\\1}}$$

for order 4

$$D^*_{1111} = \frac{1}{N} C^*_1 - \frac{4}{N^2} C^*_2 + \frac{6}{N^3} C^*_3 - \frac{3}{N^4} C^*_4$$

$$D^*_{1112} = \frac{-1}{N^2} C^*_{\substack{1\\1}} + \frac{3}{N^3} C^*_{\substack{2\\1}} + \frac{3}{N^4} C^*_{\substack{3\\1}}$$

$$D^*_{1122} = \frac{2}{N^3} C^*_{\substack{2\\1}} - \frac{3}{N^4} C^*_{\substack{2\\2}}$$

$$D^*_{1123} = \frac{1}{N^3} C^*_{\substack{1\\1\\1}} - \frac{3}{N^4} C^*_{\substack{2\\1\\1}}$$

$$D^*_{1234} = \frac{3}{N^4} C^*_{\substack{1\\1\\1\\1}}$$

for order 5

$$D^*_{11111} = \frac{1}{N} C^*_1 - 5\frac{1}{N^2} C^*_2 + \frac{10}{N^3} C^*_3 - \frac{5}{N^4} C^*_4 + \frac{4}{N^5} C^*_5$$

$$D^*_{11112} = \frac{-1}{N^2} C^*_{\substack{1\\1}} + \frac{4}{N^3} C^*_{\substack{2\\1}} - \frac{6}{N^4} C^*_{\substack{3\\1}} - \frac{4}{N^5} C^*_{\substack{4\\1}}$$

$$D^*_{11122} = \frac{1}{N^3} C^*_{\substack{2\\1}} - \frac{1}{N^4} C^*_{\substack{3\\1}} - \frac{2}{N^4} C^*_{\substack{2\\2}} + \frac{4}{N^5} C^*_{\substack{3\\2}}$$

$$D^*_{11123} = \frac{1}{N^3} C^*_{\substack{1\\1\\1}} - \frac{3}{N^4} C^*_{\substack{2\\1\\1}} + \frac{4}{N^5} C^*_{\substack{3\\1\\1}}$$

$$D^*_{11223} = -\frac{2}{N^4} C^*_{\substack{2\\1}} + \frac{4}{N^5} C^*_{\substack{2\\2}}$$

$$D^*_{11234} = -\frac{1}{N^4} C^*_{\substack{1\\1\\1}} - \frac{4}{N^5} C^*_{\substack{2\\1\\1}}$$

$$D^*_{12345} = \frac{4}{N^5} C^*_{\substack{1\\1\\1\\1\\1}}$$

6. D^*_U for Finite Version of Cumulants $\kappa_{111\ldots1}$

Unbiased estimates of finite version of the cumulants are considered here as another application of unbiased estimates of the gmfp for sampling without replacement for finite population (Mikhail and Malik, 1978).

Here the coefficient B_U for U = 111...1 is

$$B_U = (-1)^{n-1}(n-1)!/N^n \quad \text{if the number of rows is } n.$$

Since all moment functions are identical for order 1, 2, and 3, we can say that unbiased estimates of moment functions of the population are identical for order $p \le 3$. Here we start for cumulants of order $p \ge 4$.

for order 4

$$D^*_{1111} = \frac{1}{N} C^*_1 - \frac{4}{N^2} C^*_2 - \frac{3}{N^2} C^*_2 + \frac{12}{N^3} C^*_3 - \frac{6}{N^4} C^*_4$$

$$D^*_{1112} = -\frac{1}{N^2} C^*_{\substack{1\\1}} + \frac{6}{N^3} C^*_{\substack{2\\1}} - \frac{6}{N^4} C^*_{\substack{3\\1}}$$

$$D^*_{1122} = \frac{-1}{N^2} C^*_{\substack{1\\1}} + \frac{4}{N^3} C^*_{\substack{2\\1}} - \frac{6}{N^4} C^*_{\substack{2\\2}}$$

$$D^*_{1123} = \frac{2}{N^3} C^*_{\substack{1\\1\\1}} - \frac{6}{N^4} C^*_{\substack{2\\1\\1}}$$

$$D^*_{1234} = -\frac{6}{N^4} C^*_{\substack{1\\1\\1\\1}}$$

for order 5

$$D^*_{11111} = \frac{1}{N} C^*_1 - \frac{5}{N^2} C^*_2 - \frac{10}{N^2} C^*_2 + \frac{20}{N^3} C^*_3 + \frac{30}{N^3} C^*_3 - \frac{60}{N^4} C^*_4 + \frac{24}{N^5} C^*_5$$

$$D^*_{11112} = -\frac{5}{N^2} C^*_{\substack{1\\1}} + \frac{12}{N^3} C^*_{\substack{2\\1}} + \frac{6}{N^3} C^*_{\substack{2\\1}} + \frac{36}{N^4} C^*_{\substack{3\\1}} + \frac{24}{N^5} C^*_{\substack{4\\1}}$$

$$D^*_{11122} = -\frac{1}{N^2} C^*_{\substack{1\\1}} + \frac{8}{N^3} C^*_{\substack{2\\1}} - \frac{18}{N^4} C^*_{\substack{2\\2}} - \frac{6}{N^4} C^*_{\substack{3\\1}} + \frac{24}{N^5} C^*_{\substack{3\\2}}$$

$$D^*_{11123} = \frac{2}{N^3} C^*_{\substack{1\\1\\1}} - \frac{12}{N^4} C^*_{\substack{2\\1\\1}} + \frac{24}{N^5} C^*_{\substack{3\\1\\1}}$$

$$D^*_{11223} = \frac{2}{N^3} C^*_{\substack{1\\1\\1}} - \frac{12}{N^4} C^*_{\substack{2\\1\\1}} + \frac{24}{N^5} C^*_{\substack{2\\2\\1}}$$

$$D^*_{11234} = -\frac{6}{N^4} C^*_{\substack{1\\1\\1\\1}} + \frac{24}{N^5} C^*_{\substack{2\\1\\1\\1}}$$

$$D^*_{12345} = \frac{24}{N^5} C^*_{\substack{1\\1\\1\\1\\1}}$$

7. Applications and Summary

In this section examples of the unbiased estimates of the trivariate moment function $E^*_N \mu\begin{pmatrix}1\\1\\1\end{pmatrix}$ and cumulants $E^*_N \kappa\begin{pmatrix}1\\1\\1\end{pmatrix}$ are obtained as applications to the unbiased estimates of the multivariate gmfp $E^*_N F\left\{\begin{pmatrix}1\\1\\1\end{pmatrix}\right\}$ in sections 5 and 6 using the D^*_U-functions in sections 2 and 3. For example,

for $\mu\begin{pmatrix}1\\1\\1\end{pmatrix}$

$$E^*_N \mu\begin{pmatrix}1\\1\\1\end{pmatrix} = (111)\left(\frac{1}{N} C^*_1 - \frac{3}{N^2} C^*_2 + \frac{2}{N^3} C^*_3\right) + ((112) + (121) + (211)) \left(\frac{-1}{N^2} C^*_{\substack{1\\1}} + \frac{2}{N^3} C^*_{\substack{2\\1}}\right) + (123) \frac{2}{N^3} C^*_{\substack{1\\1\\1}}$$

for $\kappa\begin{pmatrix}1\\1\\1\end{pmatrix}$

$$E^*_N \kappa\begin{pmatrix}1\\1\\1\end{pmatrix} = (111)\left(\frac{1}{N} C^*_1 - \frac{3}{N^2} C^*_2 + \frac{2}{N^3} C^*_3\right) + ((112) + (121) + (211)) \left(\frac{-1}{N^2} C^*_{\substack{1\\1}} + \frac{2}{N^3} C^*_{\substack{2\\1}}\right) + (123) \frac{2}{N^3} C^*_{\substack{1\\1\\1}}$$

For the trivariate moment function we have

$$E^*_N(F_{111}) = D^*_{111}(111) + D^*_{110}\left(\begin{smallmatrix}110\\001\end{smallmatrix}\right) + D^*_{101}\left(\begin{smallmatrix}101\\010\end{smallmatrix}\right) + D^*_{011}\left(\begin{smallmatrix}011\\100\end{smallmatrix}\right) + D^*_{100}\left(\begin{smallmatrix}100\\010\\001\end{smallmatrix}\right).$$

For the bivariate case we get

$$E^*_N(F_{21}) = D^*_{21}(21) + D^*_{20}\left(\begin{smallmatrix}20\\01\end{smallmatrix}\right) + 2D^*_{11}\left(\begin{smallmatrix}11\\10\end{smallmatrix}\right) + D^*_{10}\left(\begin{smallmatrix}10\\10\\01\end{smallmatrix}\right),$$

and for the univariate case we have

$$E^*_N(F_3) = D^*_3(3) + 3D^*_2\left(\begin{smallmatrix}2\\1\end{smallmatrix}\right) + D^*_1\begin{pmatrix}1\\1\\1\end{pmatrix}$$

In general this paper gives the functional forms of the multivariate gmfp and its unbiased estimates in a very compact form. The results are applied to obtain multivariate unbiased estimates of moment functions $\mu_{111\ldots1}$ and cumulants $\kappa_{111\ldots1}$.

REFERENCES

Carver, H. C. (1930). Fundamentals of the theory of sampling. Ann. Math. Statist., 1, 101-121, 260-274.

Dwyer, P. S. (1937). Moments of any rational integral isobaric sample moment function. Ann. Math. Statist., 8, 21-65.

Dwyer, P. S. (1938). Combined expansions of products of symmetric power sums and sums of symmetric power products with application to sampling. Ann. Math. Statist., 9, 1-47, 97-132.

Dwyer, P. S., Mikhail, N. N. and Tracy, D. S. (1978). A concise proof of a theorem on products of power sums. Canad. J. Statist., 6, 1, 11-17.

Dwyer, P. S. and Tracy, D. S. (1964). A combinatorial method for products of two polykays with some general formulae. Ann. Math. Statist., 35, 1174-1185.

Dwyer, P. S. and Tracy, D. S. (1980). Expectation and estimation of product moments in sampling from finite population. J.A.S.A., 75, 431-37.

Mikhail, N. N. and Malik, H. J. (1978). Unbiased estimates of cumulants and products of cumulants for finite sampling. J. Nat. Sci. Math., 18, 2, 7-17.

Mikhail, N. N. and Malik, H. J. (1978). Unbiased estimates of Dwyer's moment function and their products for finite sampling. J. Indian Statist. Assoc., 16, 71-82.

Mikhail, N. N., McCall, T. M., Works, R. L. and Gillette, B. N. S. (1985). Multivariate moments of h-statistics and their products for a finite population. J. Statist. Studies, 5, 1-34.

Tracy, D. S. and Dwyer, P. S. (1973). Partitional functions of sample partitional functions. Technical Report No. 17, Dept. of Stat., Univ. of California, Riverside.

XII. COMPUTATIONAL DISCRETE MATHEMATICS

DISCRETE STRUCTURES AND RELIABILITY COMPUTATIONS

D.E. Whited, Lincoln Laboratories
D.R. Shier, College of William and Mary
J.P. Jarvis, Clemson University

Abstract

The two-terminal reliability problem for an undirected network involves calculating the probability that two distinguished sites are connected by a path of working edges. This problem is known to be NP-hard, even for the special case of planar systems. We describe efficient data structures and algorithms for calculating the two-terminal reliability for planar networks in pseudopolynomial time; that is, the time complexity is polynomially bounded in the number of paths (or the number of cutsets). Computational experience with the algorithms is also presented.

1. Introduction

The study of the *reliability* of complex systems has interested mathematicians, statisticians, electrical engineers, and computer scientists among others (Barlow and Proschan, 1975). Its applications include such diverse areas as communication and transportation systems, electrical networks, quality control, computer design, and software validation. In particular, *network reliability* addresses the synthesis and analysis of systems that can be modeled using a network of vertices and edges. The availability of either two-way or one-way communication is reflected through the use of undirected or directed networks.

The models used in network reliability are of two types: deterministic and stochastic. In deterministic network models, a fixed network is subject to attack by an intelligent adversary. Typical reliability measures used are the connectivity, cohesiveness, and diameter of the underlying graph. Such measures tend to incorporate a worst-case point of view, by concentrating on the maximum disruption that could be inflicted on the system.

In stochastic network reliability (the focus of study here), the network components are subject to failure according to some probability model. Typically, the system under consideration is treated as a network with reliable (or perfect) vertices and unreliable edges that may assume one of two states: operational or failed. The edges are assumed to fail independently, with probabilities that are known and constant over time.

The average behavior of such a system can be studied using a variety of probabilistic measures that quantify the "connectedness" of the underlying graph resulting from edge failures. In undirected networks G, the *all-terminal* reliability R(G) refers to the probability that all vertices remain connected. The *two-terminal* reliability $R_{st}(G)$ is the probability that two specified vertices s and t are connected using operational edges of the graph. An alternative measure is the expected number of vertex pairs able to communicate. This paper will be primarily concerned with the two-terminal reliability of a network in which each edge i is assumed to operate with probability p_i.

The most fundamental method of calculating $R_{st}(G)$ uses state-space enumeration and dates back to Moore and Shannon (1956). The *state* of the network can be represented with a 0-1 vector $\delta = |\delta_1, \delta_2, \ldots, \delta_m|$ whose i-th component is 1 if edge i is operating and 0 otherwise. The probability of a given state δ is then given by

$$\Pr(\delta) = \prod_{i=1}^{m} p_i^{\delta_i} (1 - p_i)^{1-\delta_i}$$

Suppose $\mathcal{D}$ is the set of all states and the variable $I_{st}(\delta)$ equals 1 if the subgraph of operational edges indicated by δ contains an *s-t path* and 0 otherwise. An s-t path is simply a minimal set of edges whose functioning ensures that s and t are connected. Then the s-t reliability is given by

$$R_{st}(G) = \sum_{\delta \in \mathcal{D}} I_{st}(\delta) \Pr(\delta) .$$

Although conceptually simple, the state-space approach is impractical because $|\mathcal{D}| = 2^m$. Improvements to this approach can be made by focusing directly on the s-t paths $\{P_1, P_2, \ldots, P_k\}$ of G. Define E_i to be the event that all edges in path P_i operate. Then

$$R_{st}(G) = \Pr[E_1 \cup E_2 \cup \cdots \cup E_k] . \qquad (1.1)$$

The probability of each event E_i is easy to calculate by the independence assumption:

$$\Pr[E_i] = \prod_{j \in P_i} p_j .$$

The evaluation of (1.1), however, typically requires complex calculations because the events are not, in general, mutually disjoint. For example, this equation can be expanded using the inclusion-exclusion formula, but there are an exponential number of terms $(2^k - 1)$ to be considered. Thus this method of calculating $R_{st}(G)$ is *exponential* in k. This exponential behavior is not surprising in view of the fact that calculation of $R_{st}(G)$ is a mathematically difficult problem. Namely, this problem belongs to the class of NP-hard problems (Rosenthal, 1977) and thus there is unlikely to exist any efficient (i.e. polynomial-time) solution procedure.

Here we address the s-t reliability problem for the special, but important, case of planar graphs. It is known that calculating $R_{st}(G)$ is still an NP-hard problem for planar graphs (Provan, 1986). Our concern then is in providing for efficient enumeration of certain combinatorial objects (s-t paths and s-t cutsets) in planar graphs, which can then be used to calculate the s-t reliability in *pseudopolynomial* time: i.e. the work involved is polynomially bounded in the number of such objects, although it still can grow exponentially with the size of the graph. Throughout, various discrete structures will be developed both as data structures and as theoretical frameworks to implement this approach. Section 2 discusses a compact representation for planar graphs, and the next two sections present efficient algorithms for generating s-t cutsets and s-t paths in planar graphs. Section 5 shows how these generated objects can then be used to calculate network reliability for such networks.

2. Representation of Planar Graphs

An undirected graph G = (V, E) consists of a finite set V of vertices and a set E of edges whose elements are unordered pairs of vertices. The edge $e = (u,v) \in E$ is said to be *incident* with u and v, and the vertices u and v are the *end points* of e. Two vertices u and v for which $(u,v) \in E$ are called *adjacent*. The set of vertices adjacent to v is written A(v), with the *degree* of v defined as |A(v)|. Throughout, we will reserve n for |V| and m for |E|.

In particular, we will be concerned with planar graphs. An undirected graph is *planar* if it can be embedded in the plane so that edges intersect only at a vertex with which they are both incident. Given an embedding of G in the plane, a *region* of G

is a maximal connected portion of the plane which does not contain elements of G. Every embedding of G in the plane has one infinite region called the *exterior* region. A planar graph will in general have many different plane embeddings, though the total number of regions r_G will always equal $m - n + 2$. Figure 2.1 provides an example of a graph and one particular embedding of it in the plane; the regions r_i have also been indicated with the exterior region called r_1

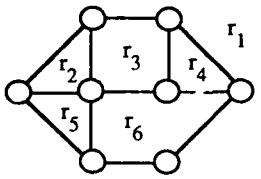


Figure 2.1 A planar graph with regions r_1 through r_6

Let G be a planar graph with a fixed plane embedding. A *dual* of G, denoted G^D, is a graph formed by associating a vertex of G^D with each region of G and then joining two vertices of G^D for each edge of G common to the boundaries of the two associated regions of G. (See Figure 2.2.) When it is clear from the context, a dual relative to a specific embedding of G will be referred to as the dual of G. The dual graph G^D is also a planar graph. Moreover, the regions of G are in one-to-one correspondence with the vertices of G^D, vertices of G with the regions of G^D, and edges of G with the edges of G^D. Note that an embedding of G^D is determined by the chosen embedding of G.

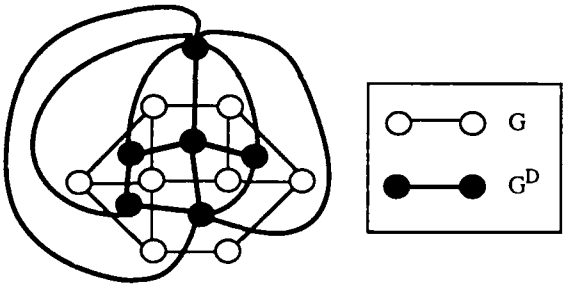


Figure 2.2 A graph G and its dual G^D

Subsequently, it will be necessary to identify the regions and edges "around" a given vertex v, relative to a fixed embedding of G. A region is incident with v whenever v is on the boundary of that region. The regions and edges incident with v can be placed in an ordered (circular) list: $r_0, e_0, r_1, e_1, \ldots, r_{d-1}, e_{d-1}, r_0$ where d is the degree of v and region r_i is bordered by e_{i-1} and e_i (all subscripts being taken modulo d). Such an ordered list will reflect either a clockwise (CW) or counterclockwise (CCW) traversal of the regions and edges incident with v. Similarly, a CW or CCW orientation can be given to a region r, inducing an ordered circular list of the vertices and edges on its boundary: $v_0, e_0, v_1, e_1, \ldots, v_{k-1}, e_{k-1}, v_0$. Here edge e_{i-1} joins vertices v_{i-1} and v_i (modulo k) for $1 \le i \le k$, with k denoting the size of the boundary of r (equivalently the degree of the dual vertex corresponding to r in the dual embedding).

Planar graphs can be encoded in a compact way for use in reliability analysis (Whited, 1986). In particular, the data structure allows easy access to (a) an ordered list of the edges and regions incident with a given vertex v; (b) an ordered list of the edges and vertices on the boundary of a given region r; (c) the two regions bordered by each edge; and (d) the corresponding representation for the dual graph. Linear time algorithms to carry out each of these four tasks can be implemented using such a data structure.

3. Enumeration of s-t Cutsets in Planar Graphs

A fundamental notion in reliability calculations is that of a *path*: a minimal set of components whose operation ensures that the system operates. Another important concept is that of a *cutset*: a minimal set of components whose failure ensures that the system must fail. We first describe such concepts in the context of graphs, and then discuss methods for enumerating these objects. For planar graphs, certain "local" information can be exploited to provide an improved cutset enumeration algorithm. Section 4 presents similar methods for the enumeration of paths. First, we establish some needed notation.

In a graph $G = (V, E)$, the complement of $X \subseteq V$ is denoted by $\overline{X} = V - X$. The *open neighborhood* $\Gamma(X)$ of X is defined by $\Gamma(X) = \{v \in \overline{X} \mid (u,v) \in E \text{ for some } u \in X\}$. The *induced subgraph* $\langle X \rangle$ is the graph $H = (X, F)$ where $F = \{(u,v) \in E \mid u,v \in X\}$.

An alternating sequence $u = v_0, (v_0,v_1), v_1, \ldots, v_{k-1}, (v_{k-1},v_k), v_k = v$ of distinct vertices and edges is called a *u-v path*. If a u-v path exists in G between all vertices u and v, then G is *connected*. Otherwise, G decomposes into a number of *connected components*. A vertex v of a connected graph is called a *cut vertex* if the graph $G - v = \langle V - v \rangle$ is not connected. A minimal set of edges whose removal from G leaves s and t in different connected components is an *s-t cutset*.

If $X \subset V$ with $s \in X$ and $t \in \overline{X}$, then $\langle X, \overline{X} \rangle$ denotes the set of edges in E with one end point in X and the other in $\overline{X}$. Note that the removal of the edges in $\langle X, \overline{X} \rangle$ separates vertex s from vertex t. If both induced subgraphs $\langle X \rangle$ and $\langle \overline{X} \rangle$ are connected, then it is known that $\langle X, \overline{X} \rangle$ is an s-t cutset (Bellmore and Jensen, 1970). For this reason, such a set X (with $\langle X \rangle$ and $\langle \overline{X} \rangle$ being connected) will be called a *connected s-t separating set*.

The most efficient algorithm for enumerating all s-t cutsets in an undirected graph G is the procedure of Tsukiyama, et al. (1980). Its worst-case time complexity is given by $O((n+m)c_{st})$, where c_{st} is the number of s-t cutsets in G. This algorithm relies on two established facts:

(1) There is a one-to-one correspondence between s-t cutsets and connected s-t separating sets.
(2) Let $X \subset Y \subset V$. If both $\langle X, \overline{X} \rangle$ and $\langle Y, \overline{Y} \rangle$ are s-t cutsets, then there exists a $v \in Y - X$ so that $\langle X+v, X-v \rangle$ is an s-t cutset. Such a set X+v is called a *1-point extension* of X.

In view of (1), it is only necessary to enumerate connected s-t separating sets. The second fact then guarantees that all separating sets can be generated by considering only 1-point extensions of separating sets. This leads to the algorithm of Tsukiyama, in which each separating set is recursively processed to find its 1-point extensions.

(Any edge not on some s-t path is termed *irrelevant*. The presence of irrelevant edges may invalidate the fact that only 1-point extensions are required to find all connected s-t separating sets. For this reason, it will be supposed throughout that G is a graph without irrelevant edges. This condition can be efficiently checked, using an algorithm of Hopcroft and Tarjan (1973).)

To see how the processing works, let X be a connected s-t separating set and let $v \in \overline{X}$. The conditions for X+v to be a connected s-t separating set are then:

(a) $v \neq t$,
(b) $\langle X+v \rangle$ must be connected (so $v \in \Gamma(X)$),
(c) $\langle \overline{X}-v \rangle$ must also be connected (so v cannot be a cut vertex of $\langle \overline{X} \rangle$).

The crucial step in the Tsukiyama algorithm is determining the set W(X) comprised of all vertices v for which X+v is a 1-point extension of X. The three conditions just stated give the following description:

$$W(X) = \{v \in \overline{X} \mid v \neq t,\ v \in \Gamma(X), \text{ and } v \notin K(\overline{X})\},$$

where $K(\overline{X})$ is the set of all cut vertices of $\langle \overline{X} \rangle$.

It is important to note that determining $K(\overline{X})$, and thus $W(X)$, for each connected s-t separating set X is the most time-consuming aspect of the algorithm. A goal of this section is to develop an efficient way to determine whether or not a given vertex $v \in \overline{X}$ is in $K(\overline{X})$ for the case when G is a planar graph.

Suppose then that X is a connected s-t separating set of the planar graph G, and let v be an element of $\Gamma(X) \subseteq \overline{X}$, $v \neq t$. Normally, determining whether $v \in K(\overline{X})$ is a "global" operation in the sense that all of $\langle \overline{X} \rangle$ must be examined (e.g. by a depth-first search) to decide whether or not v is a cut vertex. However, for planar graphs a "local" check suffices, involving only the relationships among the edges and regions surrounding v. We will use C_X to denote $\langle X, \overline{X} \rangle$ and R_X to denote those regions of G which have some edge of C_X on their boundary. Also, as in Section 2, the ordered circular list of regions and edges around v will be denoted by A_v: $r_0, e_0, r_1, e_1, \ldots, r_{d-1}, e_{d-1}, r_0$. Knowledge of which edges and regions of A_v are in C_X and R_X, respectively, will provide the "local" check indicating whether or not v is a cut vertex of $\langle \overline{X} \rangle$.

Note that $v \in \Gamma(X)$ implies that at least one edge of A_v is in C_X; the connectivity of $\langle \overline{X} \rangle$ implies that at least one edge of A_v is not in C_X. Denote by $I_v(X)$ the subset of A_v containing the edges and regions in C_X and R_X. We define $I_v(X)$ to be a *contiguous* subset of A_v if for some j and k (modulo d) $I_v(X) = \{r_j, e_j, r_{j+1}, e_{j+1}, \ldots, r_k, e_k, r_{k+1}\}$. Examples of contiguous and non-contiguous subsets $I_v(X)$ of A_v are shown in Figure 3.1. The following result (Whited, 1986) establishes the key relationship to W(X).

Theorem 3.1 Let G be an undirected planar graph with a fixed plane embedding and let X be a connected s-t separating set of G. Then $v \in \Gamma(X)$ is not a cut vertex of $\langle \overline{X} \rangle$ if and only if $I_v(X)$ forms a contiguous subset of A_v.

The characterization in Theorem 3.1 yields a simple, local check to identify whether or not v is in $K(\overline{X})$, thus simplifying the determination of W(X) for planar graphs. In addition, the planarity of G can be used to establish efficient updating schemes for various sets needed to calculate W(X). The results of this section can be combined to produce a modification of the Tsukiyama procedure which enumerates all s-t cutsets in an undirected planar graph. The modified algorithm has the same worst-case complexity as that of Tsukiyama (1980), which is $O(nc_{st})$ for planar graphs.

Both the modified and original Tsukiyama algorithm have been empirically tested on various examples taken from the reliability literature with results shown in Table 3.1 (arranged roughly in order of increasing difficulty). The execution times (on an IBM 3081-K mainframe) shown in the T_1 and T_2 columns represent the total time taken to find all s-t cutsets in a given graph after the initial set-up procedures have been executed. The time required for set-up was virtually identical for both algorithms and for all four problems, and amounted to approximately .02 seconds in each instance. The results indicate that the modified algorithm yields an improvement of up to 36% over the Tsukiyama algorithm in these planar graphs.

Also, the two algorithms have been empirically compared for (p,q) *grid graphs*, consisting of p rows of q vertices connected in a rectangular grid. In addition, vertices s and t are added, with s adjacent to the p vertices in the first column of the grid and t adjacent to the p vertices in the last column of the grid. A (4,3) grid graph is pictured in Figure 3.2.

Table 3.1 Comparison of Algorithms on Four Test Examples

Source	n	m	c_{st}	T_1	T_2	%Δ
Abraham (1979)	8	12	12	.0075	.0071	5.3
Locks (1979)	9	18	72	.0411	.0294	28.5
Bailey/Kulkarni (1986)	17	25	1721	1.1518	.7609	33.9
Fishman (1986)	20	30	7376	5.5945	3.5643	36.3

T_1 and T_2 are the execution times in seconds for the original and modified Tsukiyama algorithms respectively;
%Δ is $(T_1 - T_2)/T_1 \times 100\%$

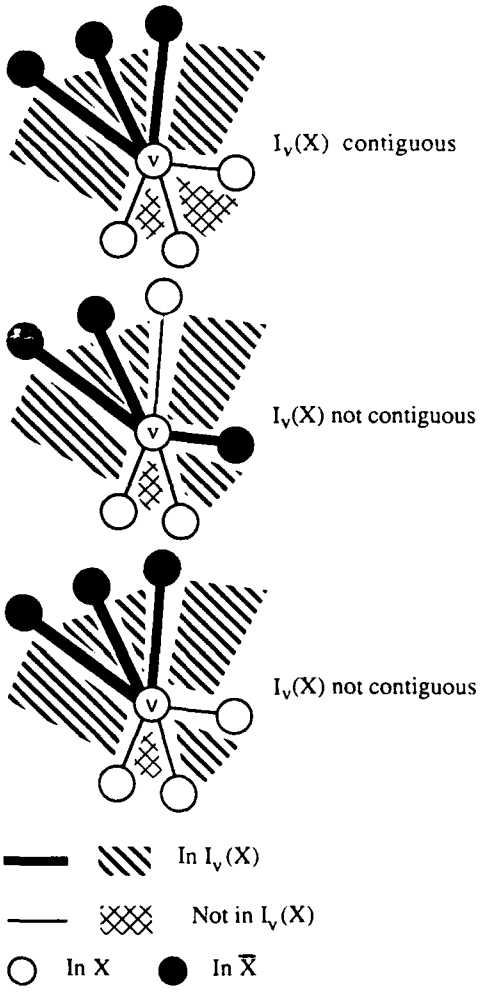


Figure 3.1 Examples of contiguous and non-contiguous $I_v(X)$

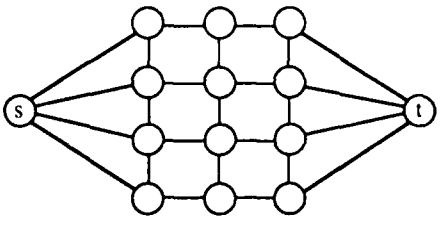


Figure 3.2 A (4,3) grid graph

The grid graphs are of interest to us for several reasons. First, they exhibit a quite rapid increase in complexity with problem size. For instance, a (3,2) grid graph has only 29 s-t cutsets, a (4,3) grid graph has 426 s-t cutsets, and a (5,4) grid graph has 16,347 s-t cutsets. Another reason for examining grid graphs is that the dual of a grid graph is also a grid graph. When the two algorithms were run on a variety of (p,q) grid graphs, the modified Tsukiyama algorithm again consistently outperformed the original procedure, with the percent improvement averaging 21%. In the most difficult problems (requiring the generation of over 100,000 cutsets), the percent improvement exceeded 40%.

4. Enumeration of s-t Paths in s-t Planar Graphs

This section describes algorithms for enumerating s-t paths in s-t planar graphs. Efficient generation of such paths is crucial for carrying out the reliability computations of the next section. An *s-t planar* graph is one which can be embedded in the plane with two specified vertices s and t lying on the boundary of the exterior region. Equivalently, G is an s-t planar graph if G together with the edge (s,t) is planar. If G is an s-t planar graph, then embed the graph $H = G + (s,t)$ in the plane and take its dual H^D. The regions of H which lie on either side of the edge (s,t) are identified with vertices s^D and t^D of H^D. The graph $G^* = H^D - (s^D,t^D)$ is then called the *s-t dual* of G. The important fact linking these two graphs is that a set of edges forms an s-t cutset (path) in G if and only if the corresponding set of dual edges forms an s-t path (cutset) in G^*.

Thus, one way to enumerate the s-t paths of G is to find its s-t dual G^* and then enumerate the s-t cutsets of G^* using the modified Tsukiyama algorithm. In fact, one can devise an alternative approach for s-t path enumeration that works directly on the given graph. The key idea again is to consider only 1-point extensions of a given path, analogous to 1-point extensions of cutsets (Section 3).

In the present case, a path P of G is associated with a set of regions R_P, and the successors of P are determined by 1-point extensions R_P+r of R_P. To define such a set of regions R_P associated with an s-t path P, notice that $C^P = P + (s,t)$ is a cycle of $H = G + (s,t)$. By the Jordan Curve theorem, the regions of H are partitioned into those "inside" C^P and those "outside" C^P. Let α be the region of H bounded by (s,t) and which is inside C^P. The set R_P consists of α and all other regions of G which are inside C^P; see Figure 4.1.

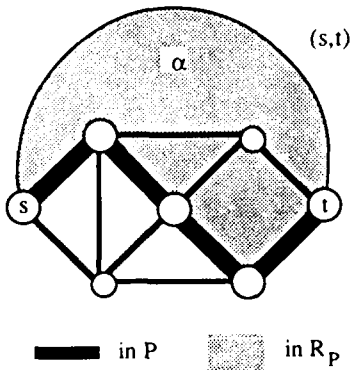


Figure 4.1 Set R_P contains the regions inside the cycle P+(s,t)

Having defined R_P, we next wish to determine all regions r such that R_P+r also corresponds in this way to an s-t path in G. Such a set R_P+r is called a *1-point extension* of R_P and the path determined by it is called a *successor* of P. The following theorem, analogous to Theorem 3.1, describes a simple "local" condition required for R_P+r to be a 1-point extension of R_P. Its proof readily follows from the identification of s-t paths of G with s-t cutsets of G^*.

Theorem 4.1 Let P be an s-t path in the s-t planar graph G. Then, R_P+r is a 1-point extension of R_P if and only if the boundary of r which lies on P forms a nontrivial subpath of P.

Figure 4.2 shows examples of regions which satisfy this requirement and of others which do not. Given a path P, its set of associated regions R_P, and a region $r \notin R_P$, we can then easily test by this theorem whether or not R_P+r forms a 1-point extension of R_P. If R_P+r is a 1-point extension, then the (new) path Q associated with R_P+r is easily derived. Namely, suppose, as in Figure 4.3, that P' is the u-w subpath of P lying on the boundary of r , where u precedes w on P. Let Q' be the other u-w path on the boundary of r. Then Q is formed from P by replacing P' with Q'. Clearly, all that is required to obtain the vertices V_Q and edges E_Q of Q from those of P is a walk of the boundary of r to identify P' and Q'.

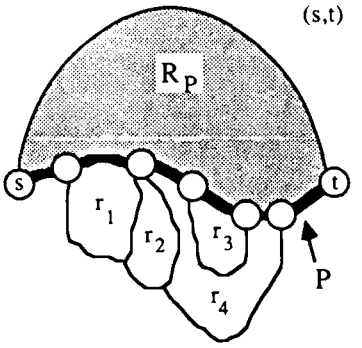


Figure 4.2 Examples of regions illustrating Theorem 4.1. r_1 and r_3 satisfy Theorem 4.1; r_2 and r_4 do not (subpath is trivial and intersection does not form subpath respectively)

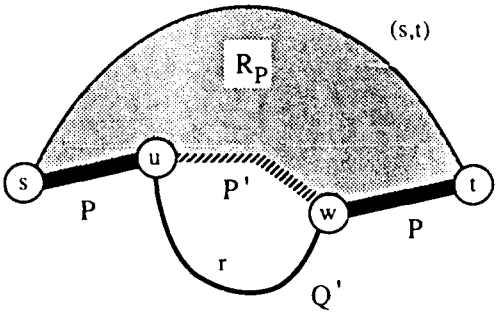


Figure 4.3 The paths P' and Q' on the boundary of r

A formal algorithm can be based on this approach, in which some improvements analogous to those presented in Section 3 are incorporated. The resulting algorithm has a worst-case time complexity of $O(np_{st})$, where p_{st} is the number of s-t paths in G. Rather than pursuing this approach in more detail we turn to a different approach that has proved to be more efficient in practice. This alternative approach to enumerating paths uses a depth-first search (DFS) of the graph. We now consider how this approach can be modified to generate s-t paths P, together with their associated regions R_P, in the presence of planarity. Although some extra work is required to find these regions, they are useful (in fact, essential) for the s-t reliability calculations presented later.

The use of a DFS to enumerate the s-t paths of a graph G requires, for each vertex v of G, the set $A(v) = \{w \in V \mid (v,w) \in E\}$ of vertices adjacent to v. Suppose that P is a current

path from s to v. We wish to extend P in all possible ways to an s-t path. The vertices in A(v) are scanned and each $w \in A(v)$ not already on P is used, in turn, to extend P to a longer path by adding (v,w) and w. The search then proceeds from each of these extensions in a recursive manner. If t is reached then a new s-t path has been found. Such a straightforward DFS procedure usually performs well in practice, but can be inefficient in the worst case. Read and Tarjan (1975) give an example of a graph with m edges and relatively few paths for which this algorithm requires on the order of 2^m steps. Read and Tarjan modified this basic DFS approach so that the recursion proceeds only when it will definitely lead to a new path. In essence, this is accomplished by looking ahead, before extending P to $w \in A(v)$, to determine whether or not this extension leads to some s-t path. This results in an $O(mp_{st})$ algorithm for a graph with p_{st} paths, which is $O(np_{st})$ for planar graphs. Subsequently it will be assumed that the basic DFS procedure has been modified in this fashion.

To aid in generating the region sets R_P at the same time, it will be convenient to use a partial ordering on the set of s-t paths. Namely, if P and Q are two s-t paths such that $R_P \subseteq R_Q$, then $P \nwarrow Q$ in the partial order. For convenience, s and t can be thought of as being on the boundary of the exterior region. Then the relation $P \nwarrow Q$ just means that P lies "above" Q (Kulkarni and Adlakha, 1985); see Figure 4.4.

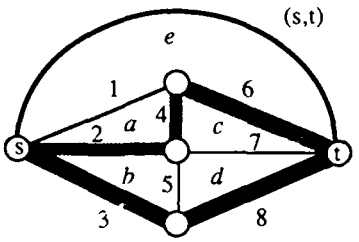


Figure 4.4 Example of two paths P and Q with $P \nwarrow Q$; P={2,4,6}, R_P={a,e}; Q={3,8}, R_Q={a,b,c,d,e}

In carrying out the DFS, the order in which paths are enumerated is determined by the order in which the vertices in each of the sets A(v) are processed. For a planar graph G, it is natural to order these sets in a manner consistent with a plane embedding of G. Specifically, we use a CW orientation around each vertex v to make A(v) an ordered circular list denoted $w_0, w_1, \ldots, w_{d-1}, w_0$. If v is reached from w_i in the DFS, then the other vertices around v will be considered in the order $w_{i+1}, w_{i+2}, \ldots, w_{i-1}$ (mod d).Several results follow from this scheme (Whited, 1986):

(1) If an s-t planar graph is embedded so that s and t lie on the border of some region α (say, the exterior region), then the search can be restricted each time a vertex lying on the boundary of α is encountered.
(2) The DFS can be performed so that P is found before Q whenever $P \nwarrow Q$.
(3) The regions R_Q can be determined easily from a knowledge of the regions R_P for each of the paths $P \nwarrow Q$.

The third result is essential to calculation of s-t reliability and hence is presented in more detail. Given any two s-t paths P and Q, define two other s-t paths as follows. The *upper path* U(P,Q) is determined by the boundary of the set of regions $R_{U(P,Q)} = R_P \cap R_Q$. Similarly, the *lower path* L(P,Q) is determined by $R_{L(P,Q)} = R_P \cup R_Q$. As seen in Figure 4.5, U(P,Q) is formed by choosing the first path in a CW traversal whenever P and Q cross, and L(P,Q) by choosing the last. Now suppose P and Q are two successive paths found by a DFS algorithm. The next theorem (Whited, 1986) shows that U(P,Q) and L(P,Q) can be used to find R_Q.

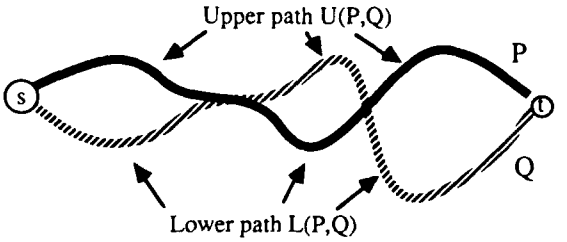


Figure 4.5 Upper path U(P,Q) and lower path L(P,Q)

Theorem 4.3 Let P and Q be two successive s-t paths found by a DFS (node adjacencies scanned CW) of an s-t planar graph G. Then for some region r of G, $R_P = R_{L(P,Q)} - r$ and $R_Q = R_{U(P,Q)} + r$.

The proof of this result relies on two facts, which can be readily established. First, whenever some portion of Q lies below P, this portion can bound only one region r. Second, only one portion Q' of Q lies below P: if Q again meets P after going below it, then Q will remain on or above P. Theorem 4.3 yields a convenient method for finding R_Q. Namely, first walk P and Q until they differ, identifying r as the region bounded from below by Q'. Next, $R_{U(P,Q)}$ is known since $R_{U(P,Q)} = R_P \cap R_Q \subseteq R_Q$ implies $U(P,Q) \nwarrow Q$ and so by our previous observation U(P,Q) is found before Q. In this way, $R_Q = R_{U(P,Q)} + r$ can be determined.

Together these three results produce an efficient DFS path enumeration procedure for undirected s-t planar graphs which not only finds all s-t paths, but also finds R_P for each path. One particular implementation issue deals with locating U(P,Q), and thus $R_{U(P,Q)}$, among the paths already generated. Since there are often many paths in even relatively small graphs, to store all paths previously generated and search the entire set would be expensive in both storage space and execution time. We can show that at most $r_G = m - n + 2$ paths need to be stored at any one time, where r_G indicates the number of regions of the planar graph G.

These paths P' (kept as a stack) are a subset of those paths satisfying $P' \nwarrow Q$, each differing from the previous by a single region in $R_{P'}$. Either the top of the stack is U(P,Q) or is removed and will never be needed again in searching for U(P,Q) (Whited, 1986). These results yield an $O(np_{st})$ algorithm for generating all paths in an s-t planar graph.

5. Pseudopolynomial Algorithms for Network Reliability

In this section, we discuss how the enumeration of s-t cutsets and s-t paths can aid in calculating $R_{st}(G)$, the probability that s and t are connected in a graph G with stochastically failing edges. Each edge i of G is assumed to operate independently with probability p_i. As discussed in Section 1, if E_i is the event that all edges in path P_i operate, then the s-t reliability is given by

$$R_{st}(G) = \Pr[E_1 \cup E_2 \cup \cdots \cup E_k] \ . \qquad (5.1)$$

In a similar way, if F_j is the event that all edges in cutset C_j fail, then the *s-t unreliability* of G is

$$U_{st}(G) = 1 - R_{st}(G) = \Pr[F_1 \cup F_2 \cup \cdots \cup F_r] \ . \qquad (5.2)$$

Various techniques, such as inclusion-exclusion, for calculating $R_{st}(G)$ or $U_{st}(G)$ require (in the worst case) an amount of work that is *exponential* in the number of objects (paths, cutsets). We would like instead a method that is polynomial in the number of objects.

Provan and Ball (1984) have shown how to calculate $U_{st}(G)$ using a certain partial order imposed on the s-t cutsets. Their algorithm is *pseudopolynomial*: namely, it has a worst-case time complexity $O(mr^2)$ which is polynomial in r, the

number of s-t cutsets of G. More generally, Shier (1988) has shown how the Provan and Ball method can be generalized and applied (for instance) to the calculation of $R_{st}(G)$ using the paths of s-t planar graphs; also see Whited (1986).

Let $E = \{e_1, ..., e_m\}$ be the set of edges and let $S = \{S_1, ..., S_r\}$ be a collection of subsets of E. (For example, the subsets might be s-t paths or s-t cutsets.) Each edge has two states, *active* and *inactive*. A set $S_i \subseteq E$ is called *active* if all its components are active. We suppose that the collection S forms a partial order $\succ$ having the *semilattice* property: namely, any two $S_i, S_j \in S$ have a unique greatest lower bound $S_i \wedge S_j$. Two additional requirements are imposed here.

(1) Closure: If S_i and S_j are active then $S_i \wedge S_j$ is active.
(2) Convexity: If $e \in S_i$ and $e \in S_j$ then $e \in S_k$ for any $S_i \succ S_k \succ S_j$.

For example, suppose we define the partial ordering $\succ$ on the s-t paths of any (s,t)-planar network as done in Section 4. Namely, $S_i \succ S_j$ if path S_i is geometrically "above" path S_j. In this case, the greatest lower bound of S_i and S_j is just $L(S_i,S_j)$, as defined earlier.

As an example of this ordering, consider the undirected graph G in Figure 5.1. The seven s-t paths and associated partial order is depicted in Figure 5.2, where each set S_i is represented by a node and the link from S_i down to S_j in the diagram represents the relation $S_i \succ S_j$. In this representation, any relations that can be inferred from the transitivity of $\succ$ are not explicitly represented by links.

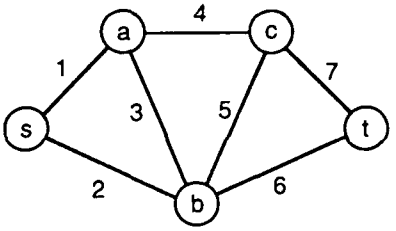


Figure 5.1 Example network

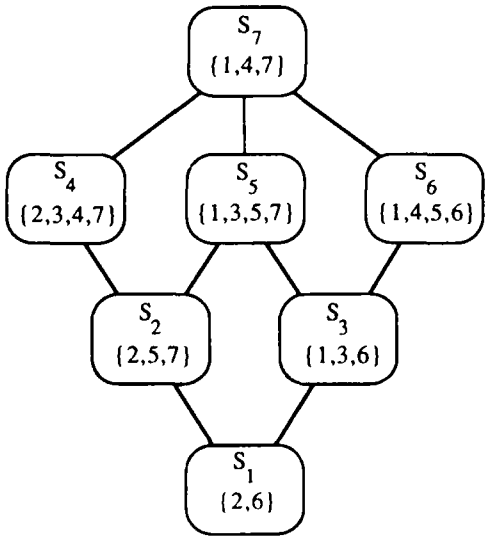


Figure 5.2 Partial ordering of s-t paths

On the other hand, we can consider the s-t cutsets of this graph. Now each such cutset S_i can be represented as $\langle X_i, \overline{X}_i \rangle$, with $s \in X_i$ and $t \in \overline{X}_i$. A natural partial ordering is then $S_i \succ S_j$ whenever $X_i \supseteq X_j$. The cutsets and associated partial order is shown in Figure 5.3.

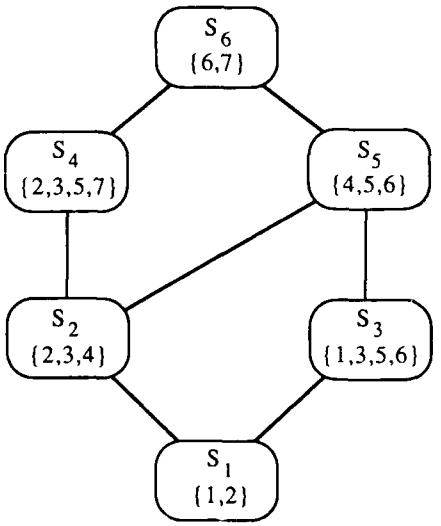


Figure 5.3 Partial ordering of s-t cutsets

We shall denote by A_i the event $\{S_i$ is active$\}$. Then our reliability calculations reduce to evaluating $\Omega(S) = Pr(A_1 \cup A_2 \cup \cdots \cup A_r)$. If we interpret "active" as meaning "functioning" and if A_i is the event that path S_i is functioning, then $\Omega(S)$ is simply $R_{st}(G)$, as seen from (5.1). If "active" means "failed" and A_i is the event that all edges in cutset S_i have failed, then $\Omega(S)$ is $U_{st}(G)$, from (5.2). It is to a general algorithm for *calculating $\Omega(S)$ that we now turn.*

Because the events A_i are not disjoint, we will instead define events L_i that are disjoint by using $L_i = \{S_i$ is the "lowest" active set in S$\}$. For example if only edge 4 fails in Figure 5.1, then S_1, S_2, S_3 and S_5 are the active sets in Figure 5.2 (none contain edge 4) and event L_1 occurs. If the sets satisfy the closure and convexity properties stated earlier, then the events L_i will indeed be a partition of the space $A_1 \cup A_2 \cup \cdots \cup A_r$. As a result, $\Omega(S)$ will equal the sum $\Sigma\, Pr(L_i)$. As shown by Shier (1988), a general recursive algorithm can be obtained that expresses $Pr(L_j)$ in terms of $Pr(A_j)$ and earlier determined $Pr(L_i)$ values:

$$Pr(L_j) = Pr(A_j) - \sum_{S_i \prec S_j} Pr(L_i)\alpha_{ij}\,, \qquad (5.3)$$

where $\alpha_{ij} = \prod Pr\{e$ is active: $e \in S_j - S_i\}$.

There are r equations represented in (5.3), each of which involves at most O(r) terms. Furthermore, each term requires at most O(m) operations to be carried out. Thus, the worst-case complexity of this method of calculating $\Omega(S)$ is $O(mr^2)$. In other words, reliability can be calculated for planar graphs in pseudopolynomial time when the s-t paths and s-t cutsets can be suitably ordered. Also, if every edge is assigned a common reliability value p, then the reliability (or unreliability) can be expressed as a polynomial in p (or in q = 1 - p) using this same method. The pseudopolynomial algorithm embodied in equation (5.3) can now be combined with the previous algorithms for efficiently generating the s-t cutsets in planar graphs or the s-t paths in s-t planar graphs.

This combined approach has enabled the calculation of reliability for some very challenging networks in the literature. We summarize the computational results for a number of (p,q) grid graphs in Table 5.1. This table lists the size of the grid networks (n vertices and m edges), the number of s-t cutsets (c_{st}), and the number of s-t paths (p_{st}). In addition, the total computation times on an IBM 3081-K mainframe are included

for calculating reliability using the s-t cutsets and also using the s-t paths. It should be emphasized that rather than simply a single numerical answer, we obtain a functional form for the reliability polynomial expressed in terms of the common edge reliability p (using the s-t paths) and for the unreliability polynomial as a function of the common edge failure probability q (using the s-t cutsets).

It is seen that there can be either more s-t cutsets or more s-t paths in such graphs, depending on the grid graph parameters. Calculation is clearly preferred using the smaller number of generated objects, and this justifies our emphasis on efficient generation of both paths and cutsets in planar graphs. Note that the (3,5) and (6,2) grid graphs are in fact duals of one another; this is manifested as the number of cutsets of one equals the number of paths of the other. Also, the (4,3) grid is self-dual: it has the same number of s-t paths as cutsets. Comparison of the associated CPU times for such (dual) grids reveals that path generation is somewhat faster than cutset generation, other things being equal.

Table 5.1 Pseudopolynomial calculation of reliability for grid graphs

p	q	n	m	c_{st}	p_{st}	T_1	T_2
2	6	14	20	49	128	.124	.400
3	3	11	18	80	95	.208	.245
3	4	14	23	195	313	1.165	2.120
3	5	17	28	444	1,033	5.834	20.724
3	6	20	33	969	3,411	28.375	197.402
4	2	10	18	95	80	.264	.196
4	3	14	25	426	426	4.464	4.103
4	4	18	32	1,745	2,320	68.975	100.724
5	2	12	23	313	195	2.358	1.116
5	3	17	32	2,320	1,745	110.134	64.005
6	2	14	28	1,033	444	22.460	5.623

T_1 is execution time in seconds, using s-t cutsets; T_2 is execution time in seconds, using s-t paths; Times do not include set-up time of approximately .02 seconds per problem

Acknowledgement. This research was supported by the U.S. Air Force Office of Scientific Research (AFSC) under Grant AFOSR-84-0154 and the Office of Naval Research (ONR) for the University Research Initiative Program under Grant N00014-86-K-0693.

References

J.A. Abraham, An improved algorithm for network reliability. *IEEE Trans. Reliability* **R-28** (1979) 58-61.

M.P. Bailey and V.G. Kulkarni, A recursive algorithm for computing exact reliability measures. *IEEE Trans. Reliability* **R-35** (1986) 36-40.

R.E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart, and Winston, New York, 1975.

M.Bellmore and P.A. Jensen, An implicit enumeration scheme for proper cut generation. *Technometrics* **12** (1970) 775-788.

G.S. Fishman, A Monte Carlo sampling plan for estimating network reliability. *Operations Research* **34** (1986) 581-594.

J. Hopcroft and R. Tarjan, Efficient algorithms for graph manipulation. *Communications of the ACM* **16** (1973) 372-378.

V.G. Kulkarni and V.G. Adlakha, Maximum flow in planar networks with exponentially distributed arc capacities. *Commun. Statist. Stochastic Models* **1** (1985) 263-289.

M.O. Locks, Evaluating the KTI Monte Carlo method for system reliability. *IEEE Trans. Reliability* **R-28** (1979) 368-372.

E.F. Moore and C.E. Shannon, Reliable circuits using less reliable relays. *J. Franklin Institute* **262** (1956) 191-208, 281-297.

J.S. Provan, The complexity of reliability computations in planar and acyclic graphs. *SIAM J. Computing* **15** (1986) 694-702.

J.S. Provan and M.O. Ball, Computing network reliability in time polynomial in the number of cuts. *Operations Research* **32** (1984) 516-526.

R.C. Read and R.E. Tarjan, Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks* **5** (1975) 237-252.

A. Rosenthal, Computing the reliability of complex networks. *SIAM J. Applied Mathematics* **32** (1977) 384-393.

D.R. Shier, Algebraic aspects of computing network reliability. Proceedings of the Third SIAM Conference on Discrete Mathematics, Clemson, S.C., (1988) 135-147.

S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi, An algorithm to enumerate all cutsets of a graph in linear time per cutset. *J. ACM* **27** (1980) 619-632.

D.E. Whited, Analysis of s-t reliability in planar graphs. Ph. D. dissertation, Clemson University, August 1986.

DETERMINING PROPERTIES OF MINIMAL SPANNING TREES BY LOCAL SAMPLING

William F. Eddy*
Carnegie-Mellon University

Allen A. McIntosh†
Bellcore

ABSTRACT

Let $\alpha_{n,k,d}$ be the fraction of vertices of degree k in a minimal spanning tree on a random sample of n vertices in d dimensions. Steele et al. (1987) show that as n increases $\alpha_{n,k,d}$ converges with probability one to an unknown constant $\alpha_{k,d}$ for any sampling distribution having a density in $\Re^d$. They perform a small scale simulation experiment to determine $\{\alpha_{k,2}, k = 1, \ldots, 5\}$ by estimating $\alpha_{n,k,2}$ for increasing values of n when vertices are distributed uniformly in the unit square. Here, we estimate $\{\alpha_{k,2}\}$ directly by systematically sampling the neighborhood of a particular vertex of the Poisson process with constant intensity in 2 dimensions. The method easily generalizes to higher dimensions. We discuss a variety of algorithms used to improve the efficiency of the sampling scheme.

1 INTRODUCTION AND SUMMARY

Let $G = (V, E)$ be a connected graph with vertex set $V = \{v\}$ and edge set $E = \{e\}$. Let $w(e)$ be a real number called the length of edge e. A *minimal spanning tree* T of G is a connected subgraph of G with vertex set V and edge set $E' \subset E$ such that

$$\sum_{e \in E'} w(e)$$

is as small as possible. From a slightly different viewpoint, a graph T is a tree if it is connected and has no circuits. A graph T is a spanning tree of a graph G if T and G have the same vertex set and T is a tree. A graph T is a minimal spanning tree (MST) of a graph G if it is the "shortest" spanning tree of G.

The work of Steele et al. (1987) demonstrated that $\alpha_{n,k,d}$, the fraction of vertices of degree k in a minimal spanning tree of the complete graph on n random vertices in d dimensions, converged with probability one to the fraction $\alpha_{k,d}$ independent of the sampling distribution. We are interested in "determining" the fractions $\alpha_{k,d}$ for $d = 2$. In Steele et al. (1987) there is some speculation concerning the possibility that the constant $\alpha_{1,2} = \frac{2}{9}$. One of our motivations for this work was to attempt to assess the validity of that speculation.

Our approach differs from the straightforward approach taken in Steele et al. (1987). They generated random samples of size n from a uniform distribution on the unit square and let n range from 16 to 65536. For each value of n only 20 minimal spanning trees were built. The total number of vertices they examined was about 2.6×10^6. Their approach suffers from two drawbacks. One drawback is the finite sample size n. The fraction $\alpha_{1,2}$ is an "asymptotic" constant. The theory derives this constant as a limit when $n \to \infty$. The second drawback is the effect of the edges of the square. It is reasonable to suppose that leaves are more frequent near the "edges" of the sample. For fixed d this effect may diminish as n increases.

The details of the theoretical derivation in Steele et al. (1987) depend on the closeness of a homogeneous planar Poisson process to a sample from a uniform distribution on the square. In fact the constants $\alpha_{k,2}$ are actually properties of the homogeneous Poisson process. Here we will generate partial realizations (subsets) of the homogeneous Poisson process and will determine the vertex degree of only one vertex v_0. The subsets only contain vertices in the vicinity of the chosen vertex. One additional benefit of our approach (which we have not taken advantage of) is that properties other than MST vertex degrees could be determined in the same way (for example, the number of Voronoi neighbors). The Voronoi polygon of each vertex of the Poisson process is that subset of the plane consisting of all points which are closer to the given vertex

*Professor of Statistics, Department of Statistics, Carnegie-Mellon University, Pittsburgh, PA 15213-3890. The work of this author was begun while he was a Resident Visitor at Bellcore, Morristown, NJ and was partially supported by the Office of Naval Research under Contract N00014-84-K-0588 and Contract N00014-87-K-0013 and National Science Foundation Grant DMS-8704218.

†Member of Technical Staff, Statistics Research Group, Bellcore, 445 South Street, Morristown, NJ 07960-1910.

than to any other vertex of the Poisson process. Two vertices of the Poisson process are Voronoi neighbors if their Voronoi polygons share a common edge.

Our approach is to generate a local piece of the Poisson realization by generating the vertices of the process which are nearest to v_0. We determine as much of the MST locally as we can, beginning at v_0. The vertex degree of v_0 in this partial MST is a lower bound on its vertex degree in the full MST of the entire Poisson process. We continue sampling and generating more of the MST until all the Voronoi neighbors of v_0 have been joined to the MST. At this point the vertex degree of v_0 is exact. This naive approach generated a new problem: the procedure often requires generation of very, very, large numbers of vertices of the Poisson process. A first revision of this approach was to "grow" the MST simultaneously from many vertices. This provided considerable improvement but was still unsatisfactory. Our second modification was to determine an upper bound on the vertex degree of v_0 by determining the full MST of the subset of the Poisson process.

In section 4 we give our estimates of the $\alpha_{k,2}$ together with some "conservative" 95% confidence intervals. These estimates are based on determining the vertex degree of approximately 1.6×10^6 vertices (but required the generation of a much larger number of vertices of the Poisson process).

2 SAMPLING A POISSON PROCESS

Let v_0 be any vertex in the homogeneous Poisson process with intensity λ in d dimensions. Let $R_1, R_2, \ldots$ be the ordered distances from v_0 to the other vertices. The joint distribution of $R_1^d, R_2^d, \ldots$ is known to be exactly the same as the distribution of the ordered distances from the origin for a homogeneous Poisson process on the line with intensity proportional to λ. See, for example, Kendall and Moran (1963). Since the homogeneous planar Poisson process is isotropic, it is easy to see that the ordered distances from a homogeneous linear Poisson process paired together with angles uniformly distributed on $[0, 2\pi)$ yield a planar process which is Poisson.

Since, in our application, we anticipate needing only those vertices of the Poisson process which are near neighbors of the chosen vertex v_0, we generate the vertices of the process in an ordered fashion. More precisely, we define an increasing sequence of sample sizes $n_0 = 1, n_1, \ldots$ and we generate an increasing sequence of circular subsets $\mathcal{V}_0, \mathcal{V}_1, \ldots$ where $\mathcal{V}_i$ contains the n_i nearest neighbors of v_0. Thus,

$$\mathcal{V}_0 = \{v_0\}$$

and

$$\mathcal{V}_0 \subset \mathcal{V}_1 \subset \ldots$$

This is not a new idea; see, for example, Quine and Watson (1984).

The advantage of this procedure in applications such as ours is obvious; it is only necessary to generate as much of the Poisson process as is necessary to determine the property of interest. Of course, it has the disadvantage, compared to the procedure in Steele et al. (1987), of requiring the generation of a great many more vertices.

3 DETERMINING VERTEX DEGREE

In this section, we develop an algorithm to find the asymptotic degree of a vertex in a given realization of the planar Poisson process. More formally, if $\mathcal{V}_0, \mathcal{V}_1, \ldots$ is a sequence of circular subsets constructed as in the previous section, with $\mathcal{V}_0 \subset \mathcal{V}_1 \subset \ldots$, and d_i is the degree of vertex v_0 in the MST of $\mathcal{V}_i$, we develop an algorithm to determine $\lim_{i \to \infty} d_i$. By running this algorithm on a large number of realizations, we may obtain $\widehat{\alpha_{k,2}} = f_k$, the observed relative frequency of vertices of degree k.

This procedure is very computationally intensive. It generates a subset of the planar Poisson process, computes a property (vertex degree) of one vertex, then throws the subset away and starts over. Why not compute the degree of more than one vertex in a given subset? This would be possible, in theory. In practice, however, we are interested in frequency estimates whose uncertainty is easily computed. If we examine one vertex from each realization, the observed frequencies have a multinomial distribution. Standard errors and confidence intervals may be computed using standard statistical theory. If we examine more than one vertex, the distribution of the observed frequencies is unknown, and we would have to verify elementary assumptions (e.g. that $\alpha_{k,2}$ is the expected value of f_k).

3.1 A Naive Algorithm

For fixed sub-sample size n, Prim (1957) shows that the MST can be built by repeated application of the following two principles:

P1: Any isolated vertex may be connected to its nearest neighbor

P2: Any tree can be connected to a nearest neighbor by the shortest available edge.

In particular, the MST may be built by applying **P1** to the vertex v_0, followed by $n-2$ applications of **P2**.

To these two principles, we add a third principle and a stopping rule:

P3: If the edge to be added by **P1** or **P2** is longer than the shortest distance from the vertex (**P1**) or tree (**P2**) to the edge of the sampling region, sample more vertices and try again.

S1: Stop when v_0 and all its Voronoi neighbors appear in the same MST.

By applying **P1** to the vertex v_0, and then applying **P2** and **P3** as often as necessary, the algorithm stops with the degree of v_0 equal to $\lim_{n\to\infty} d_n$. To see this, consider the operation of the algorithm on some subset $\mathcal{V}_k$ of vertices. Principle **P3** guarantees that the only edges added to the tree are those that will be in the MST of subsamples $\mathcal{V}_{k+1}$, $\mathcal{V}_{k+2}\ldots$ Once v_0 and its Voronoi neighbors are in the same tree, adding another edge between v_0 and one of the Voronoi neighbors will create a cycle. Since the only edges having v_0 as one endpoint must have a Voronoi neighbor as the other endpoint, we see that no further MST edges can have v_0 as an endpoint.

The performance of this algorithm is poor. Table 1 shows the results of running the algorithm on 5000 realizations of the Poisson process. In over 30 percent of these (1651/5000) the algorithm terminated after sampling 16384 vertices without determining the vertex degree of v_0.

At our talk in Reston, we showed a videotape illustrating the performance of this algorithm and the other algorithms discussed below. Figure 1 was adapted from the videotape. v_0 is the large filled circle in the center of the figure. The small filled circles represent other vertices in the MST. The edges of the MST are represented by line segments. The concentric large circles show successive circular subsets. Each subset contains twice as many vertices as the previous one. The outer circle contains 2048 vertices. The filled circle immediately to the left of v_0 is the one Voronoi neighbor not included in the MST.

Reasons for the poor performance of this algorithm are not hard to find. It is well known (see for example Bentley and Friedman 1978) that in the construction of minimal spanning trees by Prim's algorithm, trees tend to grow "uphill" towards areas of greater vertex density. If some of the Voronoi neighbors of v_0 lie across a "valley" in the vertex density, the MST may grow very large before crossing the valley. For example, in Figure 1, the unconnected Voronoi neighbor is part of a small cluster of vertices separated from the remaining vertices by a relatively large gap. The MST might cross the gap eventually; we gave up waiting after sampling 32768 vertices.

Table 1: Number of Vertices Required to Determine Vertex Degree in 5000 Realizations

Number of Vertices	Algorithm		
	Naive	Rev. 1	Rev. 2
0-64	1066	1857	4435
65-128	425	682	247
129-256	404	521	154
257-512	383	484	84
513-1024	342	322	45
1025-2048	266	251	19
2049-4096	196	235	11
4097-8192	155	148	4
8193-16384	112	106	0
Failure (> 16384)	1651	394	1

3.2 Revision 1

Based on results similar to those in Table 1, we concluded first that it was not clear that our algorithm would ever terminate in many cases, and second that even if it did terminate, it would require too much computer time. Accordingly, we set out to modify our algorithm.

By applying principle **P1** more than once, it is possible to use Prim's algorithm to grow more than one tree at a time. We modified our algorithm first to apply **P1** to all the Voronoi neighbors of v_0, and later so that it "grew" trees wherever it could (up to a limit of about 50). We felt that this might provide more opportunities to cross "valleys" in the vertex density. For example, consider the unconnected Voronoi neighbor in Figure 1. If a tree is started here, it grows to connect all the vertices in the isolated cluster. The next edge added connects this tree with the tree containing v_0. The final configuration, with a single tree containing v_0 and all its Voronoi neighbors, is shown in Figure 2. It contains only 512 vertices, a substantial improvement.

In general this extension produced considerable improvement. As Table 1 shows, our revised

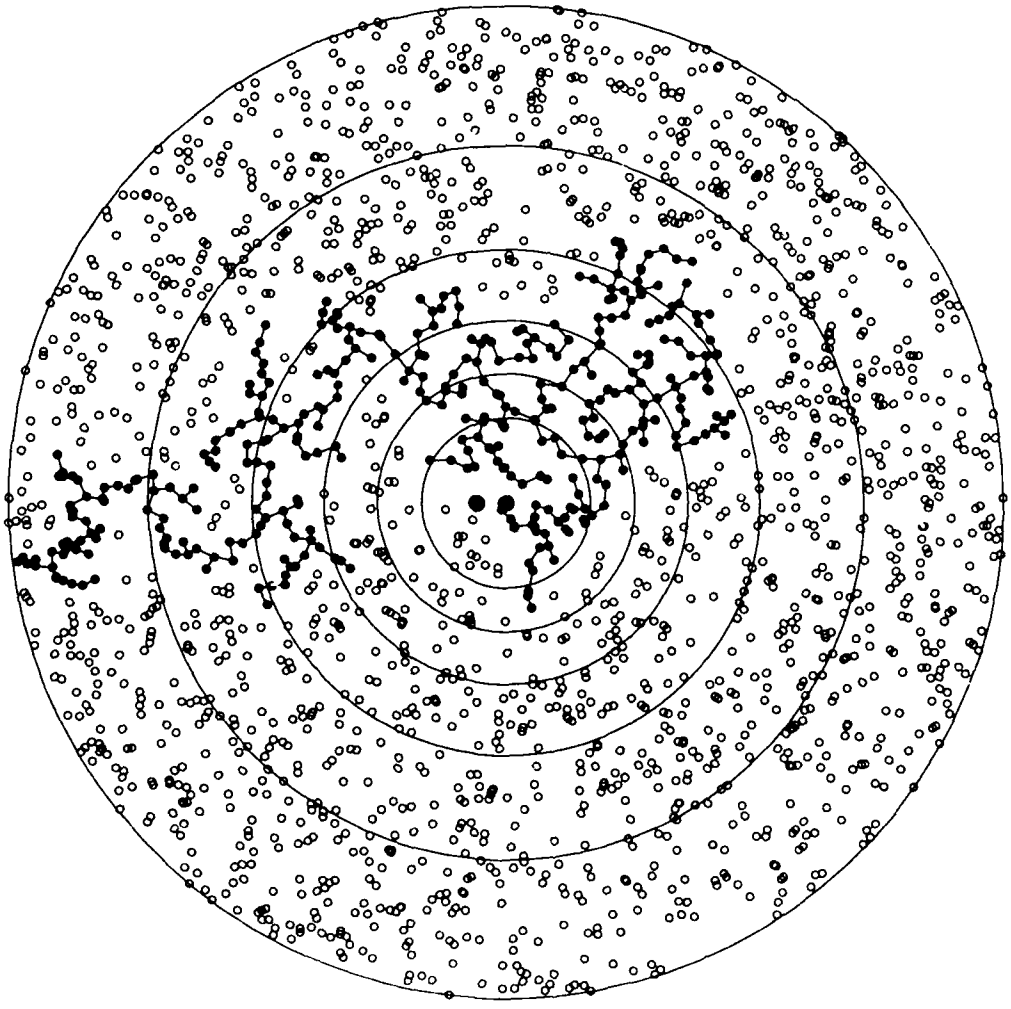


Figure 1: Naive algorithm applied to a sample of 2048 vertices

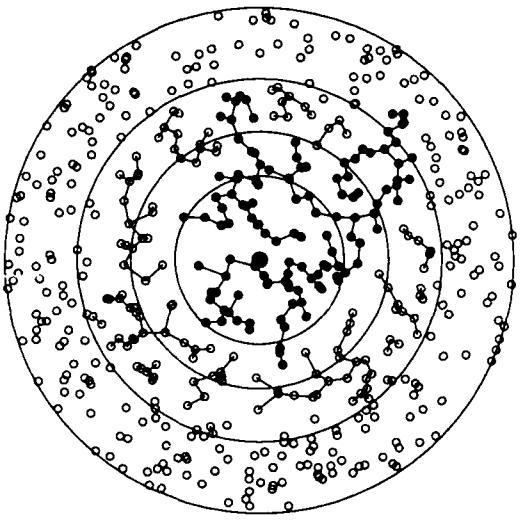


Figure 2: The algorithm, including our first revision, applied to the sample of Figure 1.

algorithm was unable to determine the vertex degree of v_0 in less than ten percent of the samples (394/5000), and in general needed to look at fewer vertices.

Unfortunately, the improvement was not enough. We conjecture that if we let this algorithm sample up to 100,000 vertices, we would be unable to determine the degree of v_0 roughly one percent of the time. With this much uncertainty, the simple confidence limits for $\alpha_{1,2}$ constructed in Section 4 would always include $\frac{2}{9}$. It might have been possible to obtain narrower confidence limits by treating the unresolved cases as censored in some fashion. We felt that we did not know enough about the censoring mechanism to make this feasible.

3.3 Revision 2

Two very important points provide us with a revised and (at long last) useful algorithm. First, as discussed above, the algorithms outlined so far compute a lower bound on the degree of v_0. Edges can be added to v_0 at any step, but can never be deleted. Second, it is possible to compute an upper bound on the degree of v_0. When the upper and lower bounds are equal, no more vertices need be sampled.

The upper bound may be obtained from the following

Lemma 3.1 *Consider the full minimal spanning trees of two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ of vertices, with $\mathcal{S}_1 \subset \mathcal{S}_2$. If $e = \{v, v'\}$ is an edge of the complete graph on $\mathcal{S}_1$ with $e \notin \mathrm{MST}(\mathcal{S}_1)$, then $e \notin \mathrm{MST}(\mathcal{S}_2)$.*

Proof This is the contrapositive of **Lemma** 2.1 of Steele et al. (1987). ■

Now consider some circular subsample $\mathcal{V}_k$ of vertices generated by the algorithm of the previous section. The algorithm provides d_k, a lower bound on the degree of v_0, and a set of trees. Suppose that no more edges can be added to these trees without sampling more vertices. Instead of doing this, the revised algorithm remembers the trees and then uses principles **P1** and **P2** to turn the trees into the full MST of $\mathcal{V}_k$. Provided that all the Voronoi neighbors of v_0 are in $\mathcal{V}_k$, Lemma 3.1 states that the degree of v_0 will never exceed the degree attained in this full MST, and hence is an upper bound on the degree of v_0 in $\mathcal{V}_{k+1}$, $\mathcal{V}_{k+2} \ldots$. If the degree of v_0 is d_k in the full MST, that is, if the degree did not change when the full MST was built, then the lower and upper bounds are equal, and the algorithm can stop. On the other hand, if edges were added to v_0, the upper bound is greater than the lower bound. In this case, the algorithm must return to the trees saved earlier apply **P3**, and continue on.

Figures 3 and 4 illustrate this algorithm in operation. The dotted lines represent edges added in building the full MST; the circular subsamples are respectively the first subsample (Figure 3) and the first two subsamples (Figure 4) from Figure 1. In Figure 3, the algorithm has started to build the MST, and has added six edges. An edge has been added to v_0. At this point, the algorithm recognizes that the upper bound on the degree of v_0 is at least two, while the lower bound is one. Construction of the rest of the full MST is pointless. In Figure 3, the algorithm has sampled more vertices, expanded the existing trees, and created some new ones. This did not produce a tree containing v_0 and all its Voronoi neighbors, so the full MST was built again. This time, the degree of v_0 did not change. Thus, the upper bound on the degree is now equal to the lower bound, and no further sampling is needed.

Table 1 shows that this revised algorithm needed 64 or fewer vertices to determine the degree of v_0 in nearly ninety percent of the samples (4435/5000). The one case requiring more than 16,384 vertices was resolved with 32,768 vertices.

4 RESULTS AND DISCUSSION

In theory, it is only necessary to sample enough vertices to make the nearest neighbor distance less than the distance to the edge of the sampling region. In practice, the nearest neighbor

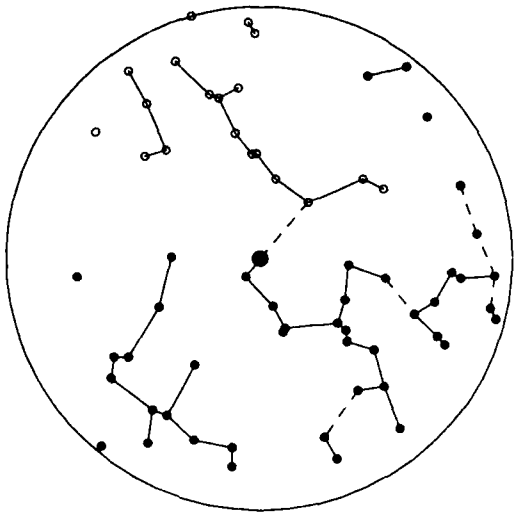


Figure 3: The final revised algorithm applied to the first subsample of Figure 1

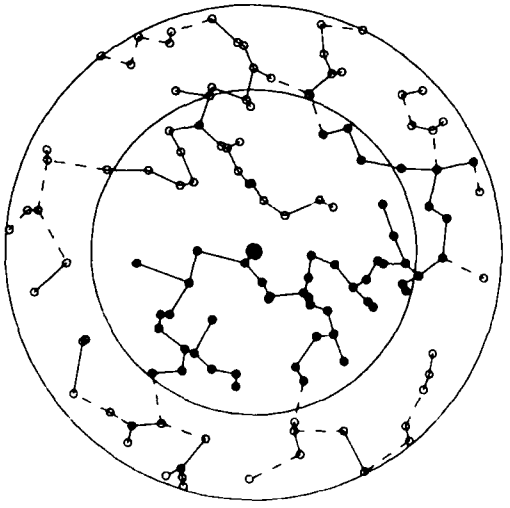


Figure 4: The final revised algorithm applied to the second subsample of Figure 1

Table 2: Observed Vertex Degree in 1,677,576 Simulation Runs

Degree k	Resolved m_k	Unresolved $m_{k,k+1}$
1	371032	60
2	948732	53
3	345270	3
4	12424	0
5	2	0

Table 3: Confidence Intervals for $\alpha_{k,2}$

Degree k	95% Confidence Interval	
1	0.220543	0.221836
2	0.564787	0.566355
3	0.205203	0.206460
4	0.007276	0.007537
5	*	*

algorithm that we use (Friedman et al. 1977) has a high setup cost. To keep the overhead cost down, we double the sample size each time we need to sample more vertices.

Bentley and Friedman (1978) suggest that the performance of their algorithm may degrade when it is used to construct the MST of a large set of vertices. We observed this behavior when building the full MST, but not when building tree fragments. This is consistent with their explanation. We found that using an algorithm due to Dwyer (1987) to build the full MST when the number of vertices was large (more than 16,384) made the simulations for large cases run two orders of magnitude faster.

4.1 Confidence Intervals

A summary of the raw figures from our simulation study is shown in Table 2. Cases that could not be resolved by sampling 131072 vertices have been tabulated according to the lower bound on the degree of v_0. In all cases the upper bound on the degree is one larger. Confidence intervals for the $\alpha_{k,2}$ are shown in Table 3. Since we observed only two vertices of degree five, we have not shown a confidence interval for $\alpha_{5,2}$. The interval for $\alpha_{1,2}$ does not lend support to the speculation that $\alpha_{1,2} = \frac{2}{9}$.

The confidence intervals were constructed as follows: Let m_k be the number of vertices whose

degree is known to be k, $m_{k,k+1}$ be the number of vertices whose degree has not been determined but is known to be k or $k+1$, and m be the total number of simulation runs. (For our data, $m = 1,677,576$, $m_1 = 371,032$, $m_{1,2} = 60$, and so on.) Since simulation runs are independent, we may view the occurrence of a vertex of degree k in a given run as a Bernoulli trial with probability $\alpha_{k,2}$. If we knew the exact vertex degree in every simulation run, we would estimate $\alpha_{k,2}$ as

$$\widehat{\alpha_{k,2}} = f_k = \frac{m_k}{m},$$

and then construct confidence intervals using the usual normal approximation. Although we do not know the exact vertex degree in every run, we may still construct conservative confidence intervals. If all vertices whose degree has not been determined but is known to be $k-1$ or k had degree k, and all vertices whose degree has not been determined but is known to be k or $k+1$ also had degree k, we would estimate $\alpha_{k,2}$ as

$$\widehat{\alpha_{k,2}} = \frac{m_k + m_{k-1,k} + m_{k,k+1}}{m}.$$

If all vertices whose degree has not been determined but is known to be $k-1$ or k had degree $k-1$, and all vertices whose degree has not been determined but is known to be k or $k+1$ had degree $k+1$, our estimate of $\alpha_{k,2}$ would be

$$\widehat{\alpha_{k,2}} = \frac{m_k}{m}.$$

In either case, the usual normal approximation may be used to construct confidence intervals. By taking the union of the two intervals formed in this way, we obtain a conservative confidence interval for $\alpha_{k,2}$.

4.2 Point Estimates

Producing point estimates is more difficult. We may write a log likelihood function for the observed counts as

$$\log \mathcal{L}(\alpha_{1,2}, \alpha_{2,2}, \alpha_{3,2}, \alpha_{4,2}, \alpha_{5,2}) = \sum_{k=1}^{5} m_k \log \alpha_{k,2} + \sum_{k=1}^{4} m_{k,k+1} \log(\alpha_{k,2} + \alpha_{k+1,2}).$$

Finding a maximum of this analytically appears to be difficult. We tried to maximize this using a fairly sophisticated function maximization routine (Gay 1983) and the initial estimates

$$\begin{aligned}
\widehat{\alpha_{1,2}} &= \frac{m_1}{m}(1 + \frac{m_{1,2}}{m_1 + m_2}) \\
\widehat{\alpha_{2,2}} &= \frac{m_2}{m}(1 + \frac{m_{1,2}}{m_1 + m_2} + \frac{m_{2,3}}{m_2 + m_3}) \\
\widehat{\alpha_{3,2}} &= \frac{m_3}{m}(1 + \frac{m_{2,3}}{m_2 + m_3} + \frac{m_{3,4}}{m_3 + m_4}) \\
\widehat{\alpha_{4,2}} &= \frac{m_4}{m}(1 + \frac{m_{3,4}}{m_3 + m_4} + \frac{m_{4,5}}{m_4 + m_5}) \\
\widehat{\alpha_{5,2}} &= \frac{m_5}{m}(1 + \frac{m_{4,5}}{m_4 + m_5}).
\end{aligned}$$

These initial estimates allocate unresolved cases based on the observed relative frequencies of resolved cases. The maximization routine was unable to do any better than the initial estimates. The values of the initial estimates are shown in Table 4.

Table 4: Estimated Values of $\alpha_{k,2}$

k	$\widehat{\alpha_{k,2}}$
1	0.221182
2	0.565586
3	0.205825
4	0.007406
5	0.000001

4.3 Discussion

The estimates in Steele et al. (1987) agree with ours to two decimal places for samples as small at 256 vertices, and to three decimal places for samples of 4096 vertices. This suggests that edge effects decrease rapidly as n becomes large. It also suggests that the constants $\alpha_{n,k,2}$ approach $\alpha_{k,2}$ at a reasonable rate. Thus, it ought to be possible to estimate $\alpha_{k,d}$ in more than two dimensions by generating vertices uniformly inside a d-cube, building the minimal spanning tree, and tabulating the observed relative frequencies of each vertex degree. Examination of the behavior of these frequencies should give some idea of how large the samples should be. If we sound cautious here, it is because intuition on this problem has been wrong in the past. Our methodology also generalizes, and hence our study could be repeated in (say) three dimensions as a check on intuition.

5 Acknowledgements

We wish to thank Joel Welling of the Pittsburgh Supercomputer Center for technical assistance in the production of our videotape.

References

Bentley, J. L. and Friedman, J. H. (1978). Fast Algorithms for Constructing Minimal Spanning Trees in Coordinate Spaces. *IEEE Transactions on Computers*, C-27(2):97–105.

Dwyer, R. (1987). A Faster divide-and-Conquer Algorithm for Constructing Delaunay Triangulations. *Algorithmica*, 2:137–151.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3):209–226.

Gay, D. M. (1983). ALGORITHM 611 - Subroutines for Unconstrained Minimization Using a Model/Trust-Region Approach. *ACM Trans. Math. Software*, 9:503–524.

Kendall, M. G. and Moran, P. A. P. (1963). *Geometrical Probability*. Griffin, London.

Prim, R. C. (1957). Shortest Connection Networks and some Generalizations. *Bell System Tech. J.*, 36:1389–1401.

Quine, M. P. and Watson, D. F. (1984). Radial Generation of n-Dimensional Poisson Processes. *J. Appl. Prob.*, 21:548–557.

Steele, J. M., Shepp, L. A., and Eddy, W. F. (1987). On the Number of Leaves of a Euclidean Minimal Spanning Tree. *J. Appl. Prob.*, 21:809–826.

MATRIX COMPLETIONS, DETERMINANTAL MAXIMIZATION AND MAXIMUM ENTROPY

Charles R. Johnson*, The College of William and Mary
Wayne W. Barrett, Brigham Young University

1. Introduction

A partial matrix is one in which some entries are (numerically) specified and the remainder are unspecified, i.e., left as free variables over some set (e.g., the field of complex numbers.) An example is

$$\begin{bmatrix} 1 & -3 & ? \\ ? & 2 & 0 \\ ? & 4 & 3 \end{bmatrix}$$

in which the "?"'s indicate unspecified entries. A completion of a partial matrix is simply a specification of the unspecified entries, resulting in a conventional matrix. For an indicated class of matrices (such as positive definite, or rank $\leq k$), the matrix completion problem is to identify partial matrices for which there is a completion in the indicated class.

Among the completion problems that have been considered are: positive definite completions [B, DGo, GJSW]; inertia possibilities [JR1]; contractions with respect to the spectral norm [JR2]; , minimum rank completions [W, JRW]; positive definite Toeplitz completions (Johnson and Rodman have been studying extensions of and a converse to the classical Caratheodory/Fejer theorem); completions of a Toeplitz contraction [JR4]; and completions which maximize the minimum eigenvalue of a partial Hermitian matrix [JR1]. Others which might be considered include stability, controllability, etc.

A feature common to many matrix completion problems is that the class of matrices of interest has an "inheritance property". Namely, all principal submatrices or all submatrices of the given matrix are in the same class. For example, all principal submatrices of a positive definite matrix are positive definite and all submatrices of a rank $\leq k$ matrix have rank $\leq k$. This imposes a necessary condition on any partial matrix that it be completable to a matrix in such a class; namely, any fully specified submatrix of the necessary sort must be in the desired class.

This raises a natural combinatorial question: Given some class of matrices, which "patterns" for the specified entries ensure an affirmative answer to the matrix completion problem, as long as specified submatrices meet the obvious necessary conditions? The next section will review the solution to the completion problem for positive definite matrices.

2. Positive Definite Case (General Theory)

We begin by defining terms and introducing notation we need to describe these results. A partial Hermitian matrix $A = (a_{ij})$ is a square partial matrix whose specified diagonal entries are real and such that if a_{ij} is specified, then so is a_{ji}, with $a_{ji} = \bar{a}_{ij}$. A partial positive definite matrix is a partial Hermitian matrix each of whose specified principal submatrices is positive definite. (By a specified portion of a partial matrix we always mean one composed entirely of specified entries.) Partial positive semidefinite matrices are defined similarly. We say that a partial positive definite (positive semidefinite) matrix is completable if it has a positive definite (positive semidefinite) completion. If A is an n-by-n partial (or full) Hermitian matrix and $\alpha \subseteq \{1, 2, \ldots, n\}$ is an index set, $A[\alpha]$ denotes the principal submatrix of A contained in the rows and columns indicated by α.

We illustrate with the simple special case

$$A = \begin{bmatrix} a_1 & b_1 & x \\ \bar{b}_1 & a_2 & b_2 \\ \bar{x} & \bar{b}_2 & a_3 \end{bmatrix} \qquad (2.1)$$

in which x is the one unspecified entry. (Throughout we refer to an unspecified pair $(x, \bar{x})$ as one unspecified entry.) The obvious necessary conditions that A have a positive definite completion are:

$$a_1 > 0, a_2 > 0, a_3 > 0,$$

$$\begin{vmatrix} a_1 & b_1 \\ \bar{b}_1 & a_1 \end{vmatrix} > 0, \begin{vmatrix} a_2 & b_2 \\ \bar{b}_2 & a_3 \end{vmatrix} > 0 \qquad (2.2).$$

By the well known criterion that a Hermitian matrix is positive definite if its leading principal minors are positive [HJ, p. 404], we see that A is completable if x can be chosen so that $\det A > 0$. It is a straightforward calculation (e.g. see equation (5.1) in [BF]) that,

$$\det A = \frac{\begin{vmatrix} a_1 & b_1 \\ \bar{b}_1 & a_2 \end{vmatrix} \cdot \begin{vmatrix} a_2 & b_2 \\ \bar{b}_2 & a_3 \end{vmatrix} - |b_1 b_2 - a_2 x|^2}{a_2}. \quad (2.3)$$

Since $a_2 > 0$, we may set $x = \frac{b_1 b_2}{a_2}$ ensuring that $\det A > 0$. Therefore A is always completable provided the necessary conditions (2.2) are met. Furthermore, the set of all x which give a positive definite completion,

$$\left\{ x : \left| x - \frac{b_1 b_2}{a_2} \right|^2 < \frac{\begin{vmatrix} a_1 & b_1 \\ \bar{b}_1 & a_2 \end{vmatrix} \cdot \begin{vmatrix} a_2 & b_2 \\ \bar{b}_2 & a_3 \end{vmatrix}}{a_2} \right\}$$

is a disc in the complex plane whose center $x = \frac{b_1 b_2}{a_2}$ gives the maximum possible determinant for a completion of A. We call the completion with $x = \frac{b_1 b_2}{a_2}$ the determinant-maximizing completion of A. Notice that setting $x = \frac{b_1 b_2}{a_2}$ is equivalent to setting the cofactors $\begin{vmatrix} b_1 & x \\ a_2 & b_2 \end{vmatrix}, \begin{vmatrix} \bar{b}_1 & a_2 \\ \bar{x} & \bar{b}_2 \end{vmatrix}$ of A equal to 0, which is the same as requiring that the 1,3 and 3,1 entries of A^{-1} be 0. These are precisely the entries in A^{-1} which correspond to unspecified entries in A.

Now suppose that A is an n-by-n partial positive definite matrix with the 1,n entry as its only unspecified entry. Applying equation (5.1) of [BF] in exactly the same manner we see that A is completable if the necessary conditions are met. If the principal submatrices $A[\{1, 2, \ldots, n-1\}]$ and $A[\{2, 3, \ldots, n\}]$ are positive definite the 1,n entry may be chosen so as to make det A (and all its principal minors) positive. The set of all values giving a positive definite completion is again a disc in the complex plane and the center gives the maximum determinant. Again the 1,n and n,1 entries of the inverse being 0 characterizes the determinant-maximizing completion. This case seems first to have been noted in [B] by different means and from the point of view of maximum entropy methods. That work, noting the connection between maximum entropy and determinant maximization, motivated interest in positive definite completion.

In [DGo] a square partial matrix is called <u>banded</u> if all the entries within some band width (symmetric from the diagonal) are specified and all entries outside are unspecified. Recall that a full matrix $A = (a_{ij})$ is called banded with band width k if $a_{ij} = 0$ whenever $|i - j| > k$. The above discussion of a single unspecified entry (whose position could be arbitrary) is just the special case of an n-by-n partial positive definite matrix with band width n - 2.

Assuming that A is an n-by-n banded partial matrix three principal conclusions are drawn in [DGo]. (The result (iii) is also contained in [BF].)

(i) positive definite completions of A necessarily exist!

(ii) There exists a unique determinant-maximizing positive definite completion.

(iii) There exists a unique completion which is nonsingular and whose inverse is banded (with the same band width) in the usual sense; this is also the determinant-maximizing completion.

We now consider the general question: for which partial Hermitian matrices do positive definite completions exist? Necessarily, the matrix must be partial positive definite, but is this condition sufficient to ensure a positive definite completion? If not, which <u>patterns</u> (in addition to banded) for the specified entries guarantee a positive definite completion?

We first note that positive definite completions need not always exist even when the obvious necessary conditions are met. It is easy to see ([GJSW]) that given a pattern for the specified entries, the matrix completion problems for positive definite and positive semidefinite matrices are equivalent. For simplicity we consider the

positive semidefinite case, and consider the partial Hermitian matrix

$$B=\begin{bmatrix} 1 & 1 & 1 & y \\ 1 & 1 & x & -1 \\ 1 & \bar{x} & 1 & 1 \\ \bar{y} & -1 & 1 & 1 \end{bmatrix} \qquad (2.4)$$

It is partial positive semidefinite as the only specified principal submatrices are

$$[1],\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},\ \text{and}\ \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

However x completes both partial principal submatrices

$$B[\{1,2,3\}]=\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & x \\ 1 & \bar{x} & 1 \end{bmatrix}$$

and

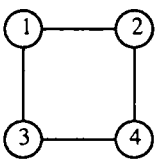
$$B[\{2,3,4\}]=\begin{bmatrix} 1 & x & -1 \\ \bar{x} & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

Since $\det B[\{1,2,3\}] = -|x-1|^2$ and $\det B[\{2,3,4\}] = -|x+1|^2$, the first submatrix requires that x = 1 for a positive semidefinite completion while the second requires that $x=-1$. As these are in conflict, B is not completable to a positive semidefinite matrix.

Of course, some partial positive definite matrices whose specified entries have the same pattern as B may have positive definite completions. Take any 4-by-4 positive definite matrix C and replace the secondary diagonal by (Hermitianly) unspecified entries. Then this partial matrix has a positive definite completion, namely C.

The interesting question then is: for which patterns does a partial positive definite matrix always have a positive definite completion. This question is addressed in [GJSW] and a characterization of completable patterns is given. A natural way of describing patterns is in terms of the undirected graph G=G(A) of the specified entries.

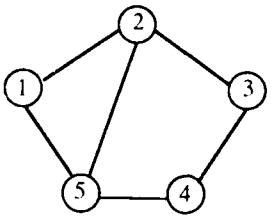
Given an n-by-n partial Hermitian matrix A, G(A) has vertex set $\{1, 2, \ldots, n\}$ and an edge between i and j, $i \neq j$ if and only if the i,j entry of A is specified. Thus, the partial matrix B above has the graph



(2.5)

Undirected graphs are appropriate because we assume the partial matrix Hermitian. Without loss of generality, from now on we assume that all diagonal entries of A are specified (because a partial positive definite matrix is completable if and only if the principal submatrix corresponding to the specified diagonal entries is completable.)

We briefly review some basic ideas about undirected graphs. A path $(i_1, i_2, \ldots, i_k)$ is a sequence of vertices such that $\{i_j, i_{j+1}\}$ is an edge of G for $j = 1, \ldots, k-1$. G is connected if there is a path between any two vertices in V. A circuit is a path for which $i_k = i_1$ and $k > 3$. A simple circuit is a circuit for which $i_1, i_2, \ldots, i_{k-1}$ are distinct. A chord of a circuit is an edge joining two nonconsecutive vertices in the circuit. A circuit is minimal if it has no chord. For example, in the graph G_1



{1, 2, 5, 4} is a path, {1, 2, 3, 4, 5, 1} is a simple circuit, {2, 5} is a chord of this circuit, and {2, 3, 4, 5, 2} is a minimal simple circuit.

The key notion which allows a simple description of completable patterns is that of a chordal graph: we call G chordal if it has no minimal simple circuits of four or more edges. Thus, the graph G_1 above is not chordal because of the minimal simple circuit {2, 3, 4, 5, 2}. However addition of the single edge {2, 4} would make it a chordal graph. A good general reference for chordal (also called triangulated) graphs is Chapter 4 of [G]. They have been heavily studied in graph theory and have arisen

before in numerical linear algebra, in the study of Gaussian elimination on sparse matrices. It is worth noting that virtually all computational tasks on chordal graphs can be carried out cheaply.

A principal result of [GJSW] is **Theorem 1**: Every partial positive definite matrix with graph G has a positive definite completion if and only if G is chordal.

A summary of the proof can be found in [J]. We simply note here that the graph of any banded partial matrix is chordal so that this theorem gives a complete generalization of conclusion (i) above from [DGo]. Note also that the graph (2.5) of the matrix B defined by equation (2.4) is the simplest non-chordal graph. According to Theorem 1, not every partial positive definite matrix with this graph is completable, and the matrix B is an example of one that is not. It is typical of a general class of counterexamples that exhibits chordality as a necessary condition for completability.

Provided that it is known there exists a positive definite completion to a partial positive definite matrix, conclusions (ii) and (iii) of [DGo] carry over irrespective of the pattern of the unspecified entries. This is another principal result in [GJSW].

Theorem 2. Suppose that the partial positive definite matrix A has a positive definite completion. Then there is a unique determinant-maximizing completion M which is also the unique completion whose inverse has zeros in the positions corresponding to unspecified entries.

3. Determinantal Maximization

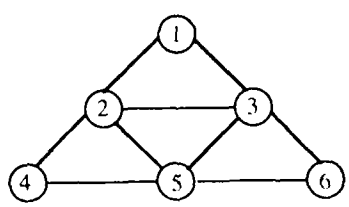
For the reasons suggested in [B] and as in [DGo] we call the determinant-maximizing completion M of a completable partial positive definite matrix A the maximum entropy completion of A. An intriguing question is: what is the value of det M as a function of the specified entries of A. The simplest example is the case in which only the diagonal entries of A are specified. If B is the completion obtained by setting all unspecified off-diagonal entries in A equal to 0, then $\det B = \prod_{i=1}^{n} a_{ii}$ and so by Hadamard's inequality, B is the determinant-maximizing completion of A. However, this simple example masks the fact that the key idea is that B^{-1} has all off-diagonal entries equal to 0.

Three distinct ways to obtain the determinant of the maximum entropy completion M of a partial positive definite matrix A have been found [BJL] whenever the graph G(A) is chordal. We summarize these here and note that the last also allows one to obtain the maximum entropy completion itself. In order to describe these results we make another digression into graph theory.

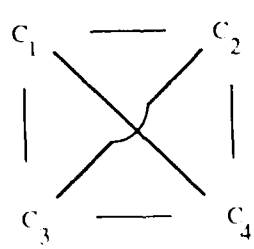
Let G be an undirected graph with vertex set $V = \{1, 2, \ldots, n\}$. A nonempty subset $C \subset V$ is called a clique of G if $\{x,y\}$ is an edge of G for all distinct $x, y \in C$. The clique C is called a maximal clique if C is not a proper subset of any clique. In the graph G_1 above, {1,2,5,}, {2,3}, {3,4} and {4,5} are the maximal cliques. Now let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be the set of maximal cliques of G. The intersection graph $G_{\mathcal{C}}$ of $\mathcal{C}$ is the graph with vertex set $\mathcal{C}$ and edge set $\mathcal{E}$ where $\{C_i, C_j\} \in \mathcal{E}$ if and only if $i \neq j$ and $C_i \cap C_j \neq \emptyset$. A subgraph T of $G_{\mathcal{C}}$ is called a spanning tree of $G_{\mathcal{C}}$ if T is a tree (a connected graph with no circuits) with vertex set $\mathcal{C}$. Such a tree T is said to satisfy the intersection property if

$C_i \cap C_j \subseteq C_k$ whenever C_k lies on the (unique) path from C_i to C_j in T (IP)

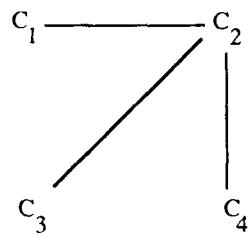
For example, let G be the graph



Then the maximal cliques are $C_1 = \{1,2,3\}$, $C_2 = \{2,3,5\}$, $C_3 = \{2,4,5\}$ and $C_4 = \{3,5,6\}$, and the intersection graph $G_{\mathcal{C}}$ is



Then the graph



is a spanning tree of G_C satisfying the intersection property (IP), while the spanning tree C_3-C_1-C_2-C_4, for example, does not satisfy (IP). The intersection property is a key hypothesis in several papers on determinantal identities and inequalities [BJ1, JB, BJ2]. Its significance in the present context is the following fundamental graph theoretic result ([BJL]).

Theorem 3. Let G be a connected undirected graph; let $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ be the set of maximal cliques of G, and let G_C be the corresponding intersection graph. Then, there is a spanning tree of G_C satisfying (IP) if and only if G is chordal.

A consequence [BJL] of Theorems 1-3 and the theorem in section 2 of [BJ2] (a formula for the determinant of a matrix based on the zero pattern of its inverse) is:

Theorem 4. Let A be a partial positive definite matrix and assume that G(A), the undirected graph of the specified entries of A, is connected and chordal. Then if B is any positive definite completion of A,

$$\det B \le \frac{\prod_{k=1}^{m} \det A[C_k]}{\prod_{\{C_i, C_j\} \in \mathcal{E}(T)} \det A[C_i \cap C_j]} \qquad (3.1)$$

where T is any spanning tree of G_C satisfying (IP) and $\mathcal{E}(T)$ is the edge set of T. Furthermore, equality is attained in (3.1) if and only if B is the maximum entropy completion of A.

Thus, the right hand side of (3.1) is a formula for the determinant of the maximum entropy completion of A in terms of its specified entries.

As a simple example, suppose that

$$A = \begin{bmatrix} 1 & 1 & ? & ? \\ 1 & 3 & 4 & ? \\ ? & 4 & 6 & 2 \\ ? & ? & 2 & 1 \end{bmatrix}$$

and that M is the maximum entropy completion of A. Then

$$\det M = \frac{\begin{vmatrix} 1 & 1 \\ 1 & 3 \end{vmatrix} \cdot \begin{vmatrix} 3 & 4 \\ 4 & 6 \end{vmatrix} \cdot \begin{vmatrix} 6 & 2 \\ 2 & 1 \end{vmatrix}}{3 \cdot 6} = \frac{4}{9}.$$

The sets $C_i \cap C_j$ corresponding to the edges $\{C_i, C_j\} \in \mathcal{E}(T)$ can be described graph theoretically, and independently of T, as the minimal vertex separators of G(A) [BJL].

We have taken G(A) to be connected in theorem 4 for convenience since the disconnected case is easily dealt with using Fischer's inequality [HJ, p. 478].

There is an alternative to the right hand side of (3.1), an "inclusion-exclusion" representation of the maximum determinant. Suppose that A is partial positive definite, G(A) is connected and chordal, $C_1, \ldots, C_m$ are the maximal cliques of G, and M is the maximum entropy completion of A. Then ([BJL])

$$\det M = \frac{\prod_{i=1}^{m} \det A[C_i] \cdot \prod_{i<j<k} \det A[C_i \cap C_j \cap C_k] \cdots}{\prod_{i<j} \det A[C_i \cap C_j] \prod_{i<j<k<l} \det A[C_i \cap C_j \cap C_k \cap C_l] \cdots}$$

After significant cancellation the right hand side may be seen to be the same as the right hand side of (3.1). However, this formulation does have the advantage of requiring only a knowledge of the maximal cliques of G(A).

A third way to obtain the maximum determinant has the added advantage that the entries of the determinant maximizing completion are directly calculated at the same time. The maximum entropy completion can be considered to be the solution to a multi-variable maximization problem. In the case that G(A) is chordal, it is shown in [GJSW] (in order to demonstrate the sufficiency in theorem 1) that there exists a sequence of chordal graphs of G_i $i = 0, \ldots, s$ such that $G_0 = G$, G_s is the complete graph and G_i is obtained from G_{i-1} by adding a single edge. (Such "chordal

orderings" of the edges missing from a chordal graph are highly nonunique.) Furthermore, at each step, there is a unique maximal clique containing the added edge. It is therefore natural to consider a sequence of one-step maximization problems in which one selects the value of the entry corresponding to the new edge to be the one that maximizes the determinant of the principal submatrix whose index set is the new maximal clique. Each of these one-step maximization problems is the case, discussed at the beginning of section 2, of picking the maximum determinant when there is only one unspecified entry. Remarkably, the matrix obtained at the end of this sequence of one-variable maximization problems is the (unique) maximum entropy completion of A [BJL, JR3], no matter which chordal ordering of the missing edges was chosen. Thus (remarkably), when G(A) is a chordal graph, a several-variable optimization problem can be solved as a sequence of (very simple) one-variable optimization problems, making it simple to obtain the determinant maximizing completion.

4. Determinantal Inequalities

There are several attractive classical inequalities for the determinant of a positive definite n-by-n matrix $A = (a_{ij})$. For example, Hadamard's inequality (1893) states that

$$\det A \leq \prod_{i=1}^{n} a_{ii}$$

Fischer's generalization (1908) is that

$$\det A \leq \det A[\alpha] \det [\alpha^c]$$

in which $\alpha \subseteq \{1, 2, \ldots, n\}$ is any index set. The further generalization

$$\det A[\alpha \cup \beta] \leq \frac{\det A[\alpha] \det A[\beta]}{\det [\alpha \cap \beta]},$$

often called "Hadamard-Fischer" has also been known for some time.

During the 1960's and 1970's a variety of additional generalizations due to Carlson, Fan and Marcus appeared. Each of these also involves a right hand side that is a ratio of principal minors of A. It is clear from theorem 4 that for any chordal graph the right hand side of (3.1) gives such an inequality for A. (As noted, the connectedness assumption upon the graph may easily be relaxed to allow inequalities such as Hadamard's as special cases.) In fact, essentially all such ratio inequalities may be deduced from these "chordal" inequalities [JB].

References

[BF] W. Barrett and P. Feinsilver, Inverses of Banded Matrices, Linear Alg. and its Applics 41(1981), 111-130.

[BJ1] W. Barrett and C. R. Johnson, Determinantal Formulae for Matrices with Sparse Matrices, Lin. Alg. and its Applics. 56(1984), 73-88.

[BJ2] W. Barrett and C. R. Johnson, Determinantal Formulae for Matrices with Sparse Inverse, II: Asymmetric Zero Patterns, Lin. Alg. and its Applics. 81(1986), 237-261.

[BJL] W. Barrett, C. R. Johnson and M. Lundquist, Determinantal Formulae for Matrix Completions Associated with Chordal Graphs, submitted.

[BJOvD] W. Barrett, C. R. Johnson, D. Olesky, and P. van den Driessche, Inherited Matrix Entries: Principal Submatrices of the Inverse, SIAD 8 (1987), 313-322.

[B] J. P. Burg, Maximum Entropy Spectral Analysis, Ph.D. Thesis (Dept. of Geophysics), Stanford University, Stanford, CA, 1975.

[DGo] H.Dym and I. Gohberg, Extensions of Band Matrices with Band Inverses, Lin. Alg. and its Applics. 36(1981), 1-24.

[EGL] R. Ellis, I. Gohberg and D. Lay, Invertible Selfadjoint Extensions of Band Matrices and their Entropy, SIAD 8(1987), 483-500.

[G] M. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980

[GJSW] R. Grone, C. R. Johnson, E. Sa' and H. Wolkowicz, Positive Definite Completions of Partial Hermitian Matrices, Lin. Alg. and its Applics. 58(1984), 109-124

[HJ] R. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.

[J] C. R. Johnson, Optimization, Matrix Inequalities, and Matrix Completions, in Operator Theory, Analytic Functions, Matrices, and Electrical Engineering, ed. by J. William Helton, CBMS Regional Conference Series in Mathematics 68, American Mathematical Society, Providence, RI, 1987.

[JB] C. R. Johnson and W. Barrett, Spanning-Tree Extensions of the Hadamard-Fischer Inequalities, Lin. Alg. and its Applics. 66(1985), 177-193.

[JR1] C. R. Johnson and L. Rodman, Inertia Possibilities for Completions of Partial Hermitian Matrices, Lin. and Multilin. Alg. 16(1984), 179-195.

[JR2] C. R. Johnson and L. Rodman, Completion of Partial Matrices to Contractions, J. Func. Anal. 69(1986), 260-267.

[JR3] C. R. Johnson and L. Rodman, Chordal Inheritance Principles and Positive Definite Completions of Partial Matrices over Function Rings, Contributions to Operator Theory and its Applications, Birkhauser 1988 (proc. of the Mesa Conference on Operator Theory and Functional Analysis), to appear.

[JR4] C. R. Johnson and L. Rodman, Completion of Toeplitz Partial Contractions, SIAM J. on Matrix Anal. and Applics. 9(1988), 159-167.

[JRW] C. R. Johnson, L. Rodman and H. Woerdeman, work in progress on general minimum rank completions.

[W] H. Woerdeman, Minimal Rank Completions for Block Matrices, Lin. Alg. and its Applics., to appear.

*The work of this author was supported in part by National Science Foundation grant DMS 87 13762 and by Office of Naval Research contract N00014-87-K-0661.

ALGORITHMS TO RECONSTRUCT A CONVEX SET FROM SAMPLE POINTS

M. Moore, École Polytechnique and McGill University; Yves Lemay, Bell Canada;
S. Archambault, École Polytechnique, Montréal

1. INTRODUCTION

It is easy to imagine situations where one has to estimate the contour of a region from partial knowledge about that region. For example, in mining exploration a geologist wants to estimate the location of an ore deposit, this from observations made at some points.

Ripley and Rasson (1977) consider a problem of that type posed by Professor D.G. Kendall. The situation is the following: given a realization of a homogeneous Poisson process of unknown intensity within an unknown compact convex set $C \subset R^2$, we want to estimate C. Conditionally on the fact that the number of observed points N is n, these points $x_1, \ldots, x_n$ are independent and uniformly distributed on C. The arguments used by Ripley and Rasson are conditional on the value of N. The proposed reconstruction consists in a dilation of the convex hull of $x_1, \ldots, x_n$ about its centroid, this dilation being such that the area of the reconstruction is (approximately) an unbiased estimation of the area of C. The procedure is affine invariant.

In the situation where one has to reconstruct an interval on the line the criterion to evaluate a procedure is clear. Indeed if the center and the length are correctly estimated the reconstruction is good. For a set in R^2 the situation is not so simple because here there is a new element, the shape of the set. The shape cannot in general be specified by a finite-dimensional parameter, so a criterion to appreciate the estimation of the shape, that would lead to some workable procedures, is not easy to find. Moore (1984) proposes to measure the precision of a reconstruction $\hat{C}$ by $m[C \Delta \hat{C}]$, the measure of the symmetric difference between C and $\hat{C}$. There exists a complete class of solutions (reconstruction rules) with respect to this loss function, however these solutions will in general be difficult to obtain.

In the problem considered by Ripley and Rasson the observations come only from the interior of C. In many situations information coming from outside of C will also be available (e.g. in the search of an ore deposit some observations will fall outside of the deposit). We will formulate here a problem allowing to incorporate this type of situations. For the problem considered a minimal sufficient statistic to reconstruct C is known (section 2). It is however difficult to find a reconstruction rule, based on this statistic, which satisfy a pertinent criterion. This is briefly explored in section 3. An alternative approach consists to formulate reconstruction algorithms based on the minimal sufficient statistic, and to evaluate them in regard to some criteria. Three such algorithms are presented in section 4. Some results of a simulation experiment designed to compare these three algorithms are reported in section 5.

2. THE PROBLEM

Let C be an unknown compact convex set in R^2. Suppose the sample points $X_1, \ldots, X_n$ (n is given but can be the value taken by a random variable) are selected independently according to a known distribution function F on R^2 whose support includes C. For each sample point it is known, in addition to its coordinates, if it is interior or exterior to C. Based on this information it is desired to reconstruct (estimate) C. Other sampling models could be considered, see De Groot and Eddy (1983).

The sample space is

$$S=\{(x_1,i_1,\ldots,x_n,i_n): x_j \in R^2,\ i_j=0 \text{ or } 1, j=1,\ldots,n\}$$

where $i_j = 1$ if the jth sample point is interior to C and $i_j = 0$ otherwise. Let H be the closed convex hull of the interior sample points and let V be the set of the vertices of H. Clearly H is a lower bound for C. An upper bound for C is given by the union, K, of all the closed convex sets Q such that $H \subseteq Q$ and for which all the x_j with $i_j = 0$ are exterior to Q. De Groot and Eddy (1983) prove that K is star-shaped from the set H, that is if $y \in K$, $z \in H$ and $u = \alpha y + (1-\alpha)z$, $0 \le \alpha \le 1$, then $u \in K$. The set K can be constructed by noting that the complement of K is

$$\bar{K} = \bigcup_{j \in E} \{y:\ y = x_j + \lambda(x_j - z),\ z \in H,\ \lambda \ge 0\}$$

where $E = \{j: i_j = 0,\ 1 \le j \le n\}$. Figure 1 illustrates the sets H and $\bar{K}$. The unknown convex set C is such that $H \subseteq C \subseteq K$ (with probability one). Let T be the set of peaks of $\bar{K}$, a peak being a sample point x_j, $j \in E$, such that if x_j is removed then $\bar{K}$ is modified. Hachtel, Meilijson and Nadas (1981), and also De Groot and Eddy (1983) in a more general setting, have shown that (V,T) is a minimal sufficient statistic for the family $\{P_C:\ C \in \mathcal{C}\}$, $\mathcal{C}$ being a class of compact convex sets included in the support of F and P_C is the probability measure induced on S by F given C. A reconstruction rule for C should be based on (V,T). It seems however difficult to find such a rule that would be easy to implement and that would satisfy an attractive criteria. This is briefly considered in the next section.

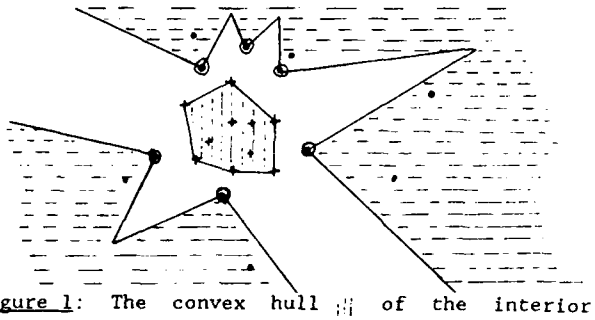


Figure 1: The convex hull H of the interior points (+); the set $\bar{K}$ [illegible] generated by the exterior points (•); the peaks are ⊙

3. RECONSTRUCTION RULES

The distribution of the random vector $(X_1, i_1, \ldots, X_n, i_n)$ given C is

$$\prod_{j=1}^{n} F(x_j) I_{C(j)}(x_j) \qquad (1)$$

where C(j) is C if $i_j = 1$ and $\bar{C}$ if $i_j = 0$, and $I_A(\cdot)$ is the indicator function of the set A. From (1) it is clear that given the observations $(x_1, i_1, \ldots, x_n, i_n)$ the maximum likelihood estimator of C is any set $\dot{C}$ such that $H \subseteq \dot{C} \subseteq K$. So the m.ℓ.e. leaves much to be chosen.

As mentionned earlier a natural measure for the accuracy of a reconstruction $\dot{C}$ is $m[C \Delta \dot{C}]$ or preferably $m[C \Delta \dot{C}]/m[C]$. Given a reconstruction rule δ, that is a function from S (more precisely $\{(V,T)\}$) to the class of the convex sets considered, a criterion to assess δ could then be

$$R(C,\delta) = E[m[C \Delta \delta(V,T)]/m[C]]$$

the expectation being with respect to the distribution defined by (1). A good reconstruction rule would be one for which $R(C,\delta)$ is minimal for a large class of sets C, or one for which the maximum value of $R(C,\delta)$ over a large class of sets C is minimal. Unfortunately, except for very restricted classes of convex sets, it will be very difficult to obtain explicitly such procedures. However, for larger classes it is sometimes possible to show that such a procedure exists.

Let λ be a (probability) measure, considered as a prior, defined on the class $\mathcal{C}$ of the convex sets considered. The posterior distribution on $\mathcal{C}$ is then given by

$$G(C|x_1,i_1,\ldots,x_n,i_n) = \begin{cases} \dfrac{\lambda(C)}{\int_{\mathcal{C}(V,T)} d\lambda(D)} & \text{if } C \in \mathcal{C}(V,T) \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where $\mathcal{C}(V,T)$ is the class of sets $C \in \mathcal{C}$ which are compatible with the observations (see De Groot and Eddy (1983)). To illustrate the reconstruction given by the mode of the posterior distribution (which maximizes $\lambda(C)$ on $\mathcal{C}(V,T)$), we consider the following example. Let $\mathcal{C}$ be the class of rectangles with sides parallel to given axes. This class can be described by the parameters (t, u, r_1, r_2) where (t, u) are the coordinates of the center and r_1, r_2 are the half lengths of the sides. As a prior we consider the measure reflecting ignorance. Following Villegas (1977) this measure (the inner prior) is such that

$$\lambda(t, u, r_1, r_2) \propto 1/r_1^2 r_2^2 \qquad (3)$$

(or $1/r_1 r_2$, the outer prior, if prior independence is assumed for the center and the lengths, but here the final procedure will be the same). The rectangle $\dot{C} \in \mathcal{C}$ maximizing (2) is the rectangle compatible with the observations and such that $r_1 r_2$ is minimal, that is the smallest rectangle including all the interior sample points. It is interesting to note that the non informative prior leads to a reconstruction using only the information provided by the interior sample points.

In some circumstances it might be possible to estimate C by the expectation of the posterior distribution (2), Hachtel, Meilijson and Nadas (1981) briefly consider this possibility.

4. ALGORITHMS

Since it is in general difficult to derive, a reconstruction rule from a general criterion, we consider some empirical algorithms. They are all based on the statistic (V,T) and are presented in order of increasing complexity.

Algorithm I(AI): The centroid o of H is determined and the maximal dilation factor, d_m, of H about its centroid, permitted by T, is determined. To find d_m we consider for each $t \in T$ the intersection u_t of the line ot with the frontier of H, then

$$d_m = \min \left\{ \frac{|ot|}{|ou_t|} , \quad t \in T \right\} .$$

The proposed reconstruction is

$$\dot{C} = [1 + \alpha(d_m - 1)]H \quad , \quad 0 \le \alpha \le 1$$

that is the dilation of H about its centroid by a factor $[1 + \alpha(d_m - 1)]$. The parameter α is chosen by the user, $\alpha = 0$ corresponds to estimate C by H and $\alpha = 1$ corresponds to the maximal dilation permitted. To reflect ignorance the value 1/2 could be assigned to α; also the choice of α could be dependent on the data. Clearly $\dot{C}$ is convex, $H \subseteq \dot{C} \subseteq K$, and the procedure is affine invariant. Figure 2 illustrates the procedure ($\alpha = 1/2$).

Algorithm II (AII): With AI the dilation is the same in all directions (isotropic). The information supplied by T may indicates some directions for which the dilation could be more important. AII takes this fact into account. Let $h_1, \ldots, h_s$ be the sides of H and $d_1, \ldots, d_s$ the maximal dilation factors permitted for each side,

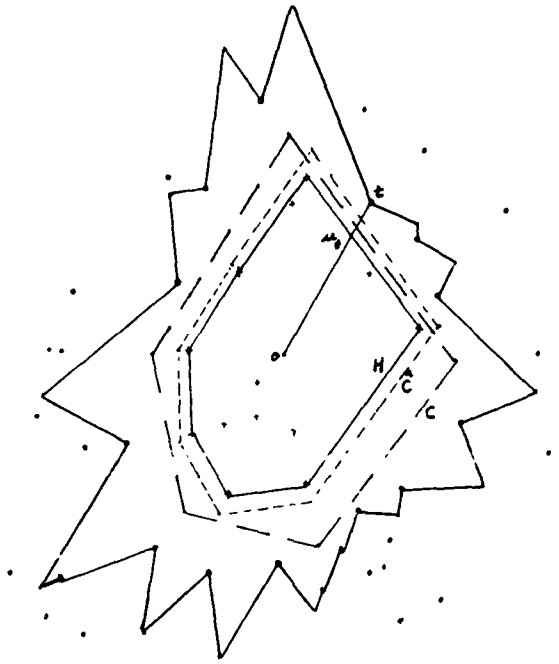


Figure 2: Algorithm I.

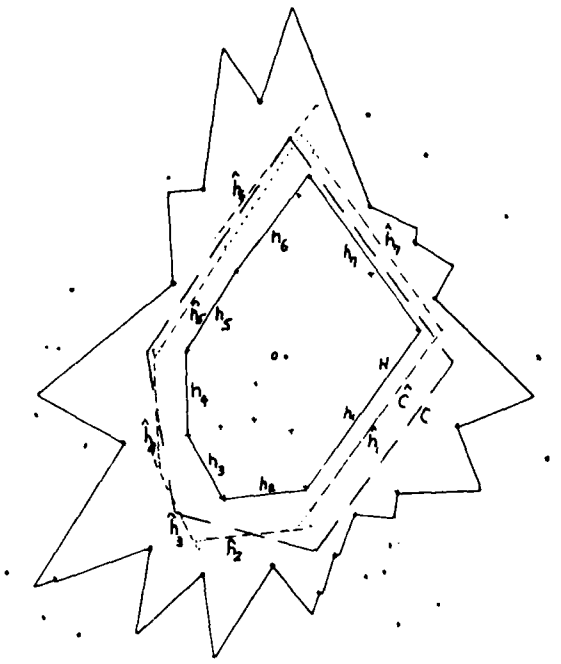


Figure 3: Algorithm II applied to the data used in Figure 2.

$$d_{mj} = \inf\left\{\frac{|ou_k|}{|ou|} : u \in h_j,\ u_k = \text{intersection of the line } ou \text{ with the frontier of } \bar{K}\right\},$$

$j = 1,\ldots,s$. Each side is dilated to become

$$\dot{h}_j = [1 + \alpha_j(d_{mj}-1)]h_j \ , \ 0 \le \alpha_j \le 1 \ , \ j = 1,\ldots,s$$

(if there are not sufficient information to determine the dilation factor for a side, then the dilation factor obtained for a neighbor side is used). The proposed reconstruction is the convex polygon obtained by extending the $\dot{h}_j$s until they meet (some $\dot{h}_j$ may be eliminated because they are too far). The role of the α_js is analogue to the one played by α for AI. By construction $\dot{C}$ is convex and $H \subseteq \dot{C} \subseteq K$. Since AII uses only dilations by factors that are invariant under affine transformations, the procedure is affine invariant. Figure 3 illustrates AII (all $\alpha_j = 1/2$) applied to the data in Figure 2, note that $\dot{h}_6$ is eliminated in the reconstruction of $\dot{C}$.

Algorithm III (AIII): With AI and AII, V is essentially used to dertermine the shape of $\dot{C}$ and T is used to fix its size. We may think this gives too much weight to V in the utilization of (V,T). The third algorithm, which is more complex, consider two preliminary estimates of C, one being simply H and the second mainly obtained from T. The final estimate is the average (Minkowski sense) of these two estimates. The hope here is that the information contained in T will be used more completely. We describe step by step (as those in the program for the simulation study) the procedure to obtain $\dot{C}$. Only the main elements are given, some complementary details, mainly about steps 5 and 8, can be found in a technical report available from the first author. Figure 4 illustrates AIII applied to the data in Figure 2.

STEP 1. Draw a frame Ω, that is a rectangle including all the sample points.

STEP 2. Let $|T|$ be the cardinality of T. If $|T| = 0$ there is no exterior point, we could then use, for example, the Ripley-Rasson procedure. If $|T| = 1$ or 2, points are possibly added to T, these are the vertices of Ω not included in $\bar{K}$ (if there are some). In the description of this algorithm T will denote the set of peaks augmented as just described. It is easy to see that T will contain at least two points.

STEP 3. Find the convex hull, W, of the points in T.

STEP 4 Determine if $H \subset W$ or not. To do so we find the number v of vertices of H which are interior to W. If $v = |V|$ then $H \subset W$ and we go to step 6 (this is the situation in Figure 4). If $v = 0$ we determine if $H \cap W \neq \phi$ or not. If this intersection is empty (this is not possible if T has been augmented in step 2) we add points to T as described in step 2 and find W from that augmented set; then $H \cap W \neq \phi$. If this new W includes H we go to step 6.

STEP 5. If $0 < v < |V|$ or if W is obtained in setp 4 and does not include H, the set W is enlarged to produce a convex set $H \subseteq \tilde{W}$ having all its vertices in K or on its frontier.

STEP 6. Let $\tilde{W}$ be the set W if $v = |V|$ or the set obtained after step 4 or 5. Consider the set $B = 1/2(H \oplus \tilde{W})$ where $\oplus$ denotes the Minkowski addition and the 1/2 contraction is with respect to the centroid of H. The set $H \oplus \tilde{W}$ is obtained by finding the convex hull of all the points in

$$((a_i+b_j): a_i \in V,\ b_j \in \tilde{T},\ i=1,\ldots,|V|,\ j=1,\ldots,|\tilde{T}|)$$

where $\tilde{T}$ is the set of vertices of $\tilde{W}$. The convex set B includes H and has all its vertices in K; however some sides of B may cross $\bar{K}$.

STEP 7. Find if some sides of B cross $\bar{K}$. To do so we determine the number u of elements in T which are interior to B. If $u = 0$ the procedure is terminated and the final reconstruction is $\hat{C} = B$ (this is the case in Figure 4).

STEP 8. If $u \geq 1$, B is reduced to form a convex set B' including H and included in K. The final reconstruction is $\hat{C} = B'$.

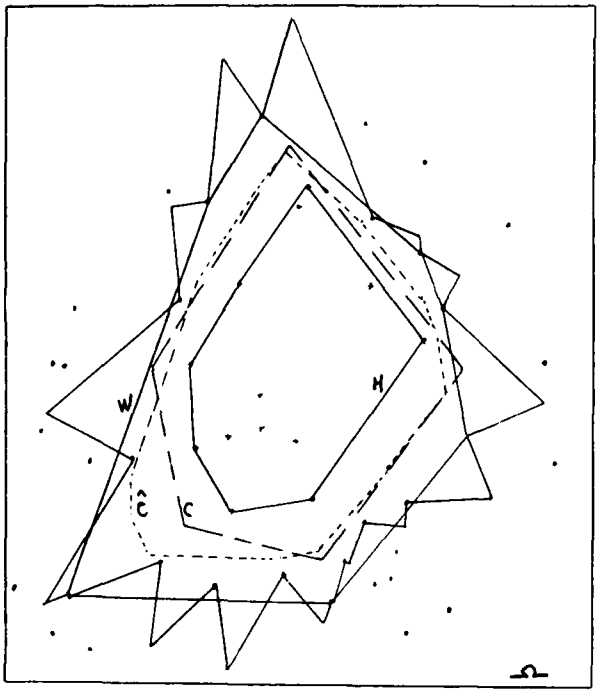


Figure 4: Algorithm III applied to the data used in Figure 2.

Because the frame Ω is introduced AIII does not really use only the information provided by the minimal sufficient statistic (V,T). However, we want to note that in many cases (if the sample size is large enough) AIII will go through steps 3, 4, 6, 7 without difficulties (e.g. Figure 4), and then the introduction of Ω (step 1) is not necessary. Also, it may happen that a set Ω is already available, it will be the case for example if the support of F is finite (see next section), or if it is known a priori that C is in a given bounded region, see Moore and Laniel (1983) for an example in soil studies.

5. A SIMULATION STUDY

To compare the algorithms presented in section 4 a simulation experiment was conducted. The structure of this experiment is the following.

A frame Ω is fixed, this is a 10 X 10 square (it will be the one used in step 1 for AIII). A polygon is drawn in Ω, this polygon is considered as the unknown C. The sample points considered are the realizations of a homogeneous planar Poisson process of intensity λ observed on Ω. To simulate these data a number n is generated from the Poisson distribution with parameter 100 λ and then n points are uniformly and independently generated on Ω. To simulate from the Poisson distribution we used the algorithm 3.3 in Ripley (1987) and to simulate from the uniform we used a procedure given by Bratley, Fox and Schrage (1983, p. 202). Each sample point is classified as interior or exterior to C. To evaluate the quality of a reconstruction two criteria are considered, a precision criterion: $m[C \Delta \hat{C}]/m[C]$, and a recovering criterion: $m[C \cap \hat{C}]/m[C]$. It is to be remarked that

$$m[C \Delta \hat{C}] = m[C] + m[\hat{C}] - 2\, m[C \cap \hat{C}]$$

so $m[C \Delta \hat{C}]$ can be large and still $m[C \cap \hat{C}]$ approximately equal to $m[C]$ i.e. C is almost recovered but with a much larger set than neccessary.

In a first stage we tried to determine the main factors influencing the results. Three factors were considered: the value of λ, the proportion of Ω occupied by C and the number of sides of C. A 2^3 factorial experiment was conducted, the selected levels were λ: 0.25, 1; proportion: 25%, 75%; number of sides: 4,12. For each of the 8 experimental conditions 250 independent repetitions were made. Relatively to both quality criteria it was observed that the first two factors are much more important than the third and that all the interactions are negligeable.

To compare the three algorithms it was decided, from the above results, to consider a polygon with six sides for C and to use three proportions: 25%, 50%, 75% and three values for λ: 0.25, 0.6, 1.0.

For each of the 27 combinations algorithm - proportion - intensity 250 repetitions were made, the 27 x 250 repetitions being independent. Tables 1 for $m[C \Delta \hat{C}]/m[C]$, and 2 for $m[C \cap \hat{C}]/m[C]$, present the average of the 250 results for each of the 27 situations. The number in parenthesis is $2s/\sqrt{250}$, s^2 being the sample variance.

Table 1 reveals that:

- For a given proportion, the precision increases with λ (i.e. when the average number of sample points increases). The relative augmentations

Table 1

Average of the 250 values of $m[C \Delta \hat{C}]/m[C]$

	Algorithm I			Algorithm II			Algorithm III		
λ / Prop.	.25	.60	1.0	.25	.60	1.0	.25	.60	1.0
25%	.582 (.022)	.352 (.015)	.261 (.011)	.514 (.022)	.293 (.014)	.204 (.008)	.335 (.011)	.281 (.010)	.215 (.008)
50%	.406 (.018)	.230 (.009)	.162 (.006)	.349 (.017)	.168 (.007)	.123 (.005)	.203 (.008)	.134 (.004)	.109 (.003)
75%	.335 (.014)	.172 (.006)	.124 (.005)	.262 (.013)	.133 (.006)	.091 (.004)	.155 (.007)	.090 (.003)	.065 (.002)

Table 2

Average of the 250 values of $m[C \cap \hat{C}]/m[C]$

	Algorithm I			Algorithm II			Algorithm III		
λ / Prop.	.25	.60	1.0	.25	.60	1.0	.25	.60	1.0
25%	.434 (.023)	.658 (.015)	.746 (.011)	.550 (.026)	.751 (.015)	.826 (.009)	.877 (.011)	.948 (.005)	.953 (.004)
50%	.604 (.018)	.775 (.010)	.841 (.006)	.676 (.019)	.855 (.007)	.895 (.005)	.866 (.011)	.944 (.004)	.964 (.003)
75%	.668 (.015)	.832 (.007)	.878 (.005)	.744 (.013)	.879 (.006)	.920 (.004)	.868 (.009)	.940 (.004)	.965 (.002)

are more important with AI and AII.

- For a given λ, the precision increases with the proportion. When the proportion is augmented and λ kept fixed, a larger proportion of the sample points is interior to C. Then H is a better approximation to C (better estimation of the shape). Also, since it is known here that C is in Ω, the precision gained from inside is not canceled by the diminution of the number of sample points outside C. The fact that AIII may use Ω explains the relatively more important increase there.
- The comparison of the algorithms by pairs, for a given proportion and a given λ, indicates that AIII does always better than AI with some important differences; AII is always better that AI; AIII is never inferior to AII and when λ is small AIII is much better than AII.
- There is an important variation among the sample variances. The variability is more important when λ or the proportion is small. The situation is similar for AI and AII but AIII appears to be more stable.

Concerning the recovering criterion, from Table 2 we observe that:

- With AI and AII, again better results are obtained when a larger proportion of the sample points are interior to C. However, AIII seems more stable in that regard.
- The comparison of the algorithm by pairs shows that AII is always better than AI and AIII always better that AII. When λ or the proportion is small AIII does much better.
- The remarks made about the variability in regard to the precision criterion also apply here.

From Table 1 and Table 2 it is easy to obtain the average of the ratio $m[\hat{C}]/m[C]$ which indicates how accurately $m[\hat{C}]$ estimates $m[C]$. We observe that AI and AII underestimate $m[C]$. This suggests that it could have been advantageous to take the α's larger than 1/2, mainly when the number of sample points is small.

To see how each of the factors: algorithm, proportion and intensity, contributes to explain the variation among the results, the ANOVA tables corresponding to a three-way layout model were computed and then the percentages of variation explained by each factor and the interactions were obtained (Table 3).

Table 3

Variation (%) explained by each factor

Criterion	Algorithm (A)	Proportion (P)	Intensity (I)	AxP	AxI	PxI	AxPxI	ERROR
$m[C \Delta \hat{C}]/m[C]$	10.5	24.0	29.5	0.0	3.0	1.0	0.0	32.0
$m[C \cap \hat{C}]/m[C]$	28.5	7.0	25.0	4.0	3.0	0.0	0.5	32.0

We remark that for each criterion the factors considered leave a large part of the variation unexplained (error). Due to the geometrical character of the problem it seems difficult to determine factors that would be easy to formulate and that would explain a larger part of the variation. Indeed, we have noticed that for a given C, a given λ, and a given algorithm, the results obtained for different samples were often very different, this being simply due ot the different position of the sample points; however, this variation was less important with AIII.

6. CONCLUSION

To reconstruct a convex set C from sample points, some being interior and the others exterior to C, a minimal sufficient statistic is known. However, a reconstruction rule based on this statistic, giving an optimal reconstruction relatively to an appealing criterion, is not in general easy to find. It is possible to formulate algorithms that are easy to apply and have acceptable performances. We have proposed three such algorithms. They are the result of many trials and partly motivated by the desire to consider simple methods. Clearly many other suggestions could be made.

When a reconstruction is evaluated two points of view can be adopted. We may be satisfied if we recover C, that is if $m[C \cap \dot{C}]/m(C)$ is near one, or we may be more severe and want that $\dot{C}$ be C, that is we want $m[C \Delta \dot{C}]$ near zero. In the first situation the choice of the algorithm is important, among those considered AIII gives good results independently of the sample size and of the relative size of C. In the second situation, since we demand much more, it seems that the sample size plays the dominant role. However, AIII gives acceptable results even for moderate sample sizes. When the sample size is important it takes much more time to apply AIII than it takes to apply AII. One may then think that the gain is not justified.

ACKNOWLEDGEMENT. This research was supported by the NSERC grant OGP0008211 and by the FCAR grant CRP-2093.

REFERENCES

Bratley, P. Fox, B.L. and Schrage, L.E. (1983). A guide to simulation. Springer Verlag, New York.

De Groot, M.H. and Eddy, W.F. (1983). Set-valued parameters and set-valued statistics. In Recent Advances in Statistics, 175-195. Academic Press, New York.

Hachtel, G.D., Meilijson, I. and Nadas, A. (1981). The estimation of a convex subset of R^k and its probability content. Technical Report RC 8666 (#37890), IBM T.J. Watson Research Center, Yorktown Heights, New York.

Moore, M. (1984). On the estimation of a convex set. Ann. Statist., 12, 1090-1099.

Moore, M. and Laniel, N. (1983). Reconstruction échantillonnale d'une forme convexe. The Canadian Journal of Statistics, 11, 181-197.

Ripley, B.D. (1987). Stochastic simulation. John Wiley, New York.

Ripley, B.D. and Rasson, J.P. (1977). Finding the edge of a Poisson forest. J. Appl. Prob. 14, 483-491.

Villegas, C. (1977). Inner statistical inference. Journal of the American Statistical Association, 72, 453-458.

APPLICATIONS OF ORTHOGONALIZATION PROCEDURES TO FITTING TREE-STRUCTURED MODELS

Cynthia O. Siu, Johns Hopkins University

Orthogonalization is an important concept in computations for linear model. In this paper, applications of Givens rotations and Modified Gram Schmidt orthogonalizations to tree-structured regressions are discussed. The resulting procedure generalizes CART's piecewise constant tree model to piecewise linear model. Great versatility is offered by this approach: regression tree models for quantitative and binary data can be handled by one general fitting procedure. In addition, it provides a basis for implementing various linear and tree-structured regression methods under one framework.

1. INTRODUCTION

Breiman et al. (1984) and Friedman (1979) described a tree-structured approach to non-parametric multiple regression. Their methods use a hierarchy of piecewise constant functions or piecewise linear functions to approximate the regression surface. For these tree models, the predictor space **X** is partitioned recursively into rectangular subregions, perpendicular to the original coordinate axes. A separate constant or linear model is fit to the subgroup of data points lying in each of the subregions obtained.

In the author's unpublished Ph.D. thesis at the University of Toronto, we propose a different tree-structured fitting procedure for piecewise linear model (Siu and Andrews, 1985). The method is based on a natural extension of Modified Gram-Schmidt Orthogonalization process in linear least squares method. Unlike the model by Friedman (1979), the hierarchy of piecewise linear models is built by adding one predictor variable at a time to the local models, linearly adjusted for the effects of those already included.

Using this orthogonalization approach, recursive partitioning is performed on residual predictor variables rather than on original variables. Data points are grouped according to a) the nature of relationships among predictor variables, and b) the relevance of these variables to the response **y**. The resulting recursive partitioning procedure is more general than the rectangular splits in previous methods. Generally, associations are found among predictor variables in real data. In the special case of having totally unrelated predictor variables, the splits performed on residual variables will be the same as the univariate rectangular splits. This method provides a simple solution to the otherwise difficult problems, such as detecting linear structures that are separated by hyperplanes not perpendicular to the coordinate axes.

In addition, this orthogonalization approach allows tree-structured models to be built and interpreted within the familiar linear regression framework. The usual selection procedures for stepwise regressions can be used for choosing variables and splitting values in this method. In the absence of recursive partitioning operations, this procedure is identical to fitting a specified linear regression by forward stepwise approach.

This paper describes applications of orthogonalization procedures to fit tree-structured models. The analogy of this approach to classical least squares methods opens many possibilities to generalize the existing tree-structured methodology. Some of them will be discussed here. In particular, the framework can be used for developing tree-structured extensions of Generalized Linear Models (GLM) (Nelder and Wedderburn, 1972). As compared to Generalized Additive Models (Hastie and Tibshirani, 1985), this recursive partitioning approach uses a hierarchy of piecewise linear functions to generalize the linear predictor function in GLM.

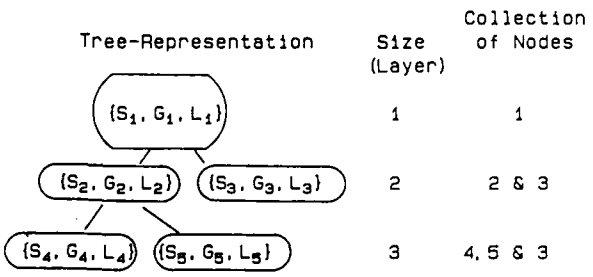
Givens rotations provide the basic algorithm for computing these tree models. The proposed fitting algorithm is flexible. It can be organized to fit a wide class of parametric and non-parametric hierarchical models within one framework. This includes as special cases stepwise procedures for generalized linear models, as well as the standard recursive partitioning procedure for piecewise constant model (Breiman et al., 1984) and piecewise linear model (Friedman, 1979). A simple model specifica-

tion rule is proposed. It helps clarify the properties and uses of these procedures.

In Section 2, we start with a simple example of the standard tree model to illustrate the basic ideas behind tree-structured methodology. Section 3 describes the linear adjustment approach to fit a tree-structured normal response model. This method differs from the one by Friedman (1979) in several important ways, will also be discussed. Section 4 shows how this approach can be used to develop the tree-structured extension of generalized linear models. Section 5 describes the model specifications for the general class of hierarchical fitting procedures considered here.

2. BINARY REGRESSION TREES

In regression tree methods, data are recursively partitioned into smaller subgroups to build a hierarchy of piecewise models. A separate local model L_k is fitted to the subgroup of data points G_k lying in each subregion S_k of the predictor space **X** (Figure 2.1).



Following the notation in Friedman (1979), each node k represents a triple (S_k, G_k, L_k) where

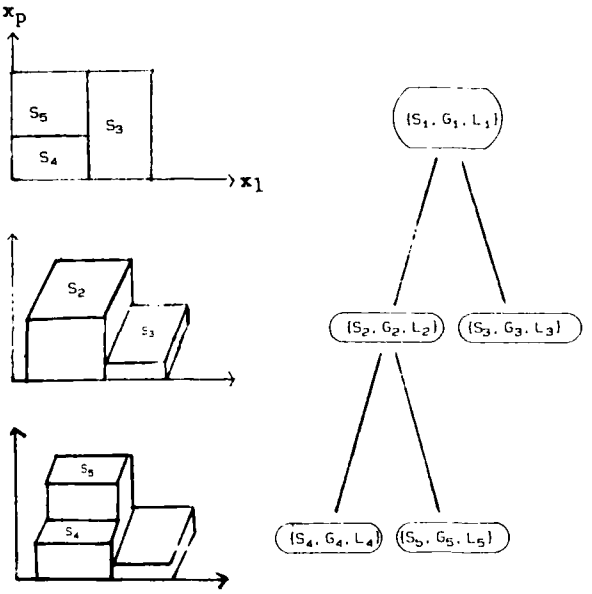
S_k = a subregion of predictor space **X**,
G_k = a subgroup of data points lying in subregion S_k, and
L_k = a local model to be applied to G_k.

Each tree model consists of layers of nested nodes, where layer is defined as a collection of nodes in one level of a tree (Siu, 1985).

Figure 2.1 illustrates a simple example of the standard tree-structured model. Starting from node 1, each nonterminal node j is split into two child nodes recursively. For each predictor variable (subscripted v), and for each value (subscripted c) of this variable, the set of data points G_k lying in subregion S_k is divided into two parts $S_{l(k)}$ and $S_{r(k)}$ - one to the left of this value and the other to the right:

A separate constant function, $L_{l(k)}=a_{l(k)}$ and $L_{r(k)}=a_{r(k)}$, is fit to the subgroups $G_{l(k)}$ and $G_{r(k)}$ at each cut point indexed by (c,v). The one (c*,k*) which best improves the fit of the model to data will be selected to split node j into two child nodes 2j and 2j+1.

Figure 2.1 Binary Regression Tree



In summary, the partitioning procedure has three components:

1) a method to divide the subregions S_k (G_k) into two parts $S_{l(k)}$ and $S_{r(k)}$ (and their corresponding $G_{l(k)}$ and $G_{r(k)}$),
2) a method to derive new local models $L_{l(k)}$ and $L_{r(k)}$ from the parent model L_k, and

3) an objective function to choose the optimal partition of parent node j.

Together, they form the basis of tree-structured methodology for regression problems. There are other important issues such as the selection of optimal-sized subtree. Different approaches have been proposed for this problem. They can be found in Sonquist and Morgan (1964), Breiman et al. (1984), Siu (1985) and Loh and Vanichsetakul (to appear in JASA).

3. FITTING REGRESSION TREE BY ORTHOGONALIZATION

3.1 Review of Linear Regression

Mosteller and Tukey (1977) gave an illuminating discussion of using linear adjustment approach to fit a specified regression by stages. The procedure is identical to Modified Gram-Schmidt Orthogonalization process in least squares method. Specifically, the method proceeds by sequentially sweeping out the effect of each predictor variable, and iteratively orthogonalizing the response y and the remaining variables to the variable swept. Let $y_{:(k)}$ be the residual of response y orthogonalized to variable subset X_k. At step k (k = 1 to p), variable x subscripted k* is added to the model $L(X_{k-1})$ by two operations:

$$\text{FIT:}\quad L(X_k) = L(X_{k-1}) + r_{k,p+1}x_{k*:(k-1)}, \qquad \text{(EQN 3.1)}$$

and

$$\text{SWEEP:}\quad x_{v:(k)} = x_{v:(k-1)} - r_{k,v}x_{k*:(k-1)}, \qquad v \neq k*.$$

The added variable plot (Cook and Weisberg, 1982) for x_{k*} is shown in Figure 3.1 (Dotted line).

3.2 Extensions to Tree-Structured Regression

To apply this linear adjustment approach to fit tree-structured piecewise linear model, the linear regressions of $y_{:(k-1)}$ and $x_{v:(k-1)}$ (v ≠ k*) on $x_{k*:(k-1)}$ in (EQN 3.1) is replaced by piecewise linear functions.

In particular, in splitting node j, the cut at (c,v) is defined by fitting a piecewise linear function $(L_{l(j)}, L_{r(j)})$ on S_j. That is,

$$L_{l(j)}(X_k) = L_j(X_{k-1}) + s_{l(j)}1 + r_{l(j)}x_{v:(k-1)}, \qquad \text{(EQN 3.2a)}$$

and

$$L_{r(j)}(X_k) = L_j(x_{k-1}) + s_{r(j)}1 + r_{r(j)}x_{v:(k-1)}$$

to $G_{l(j)}$ and $G_{r(j)}$ respectively. Let (c,v) be the cth ordered value of the residual variable subscripted v, i.e. $x_{(c),v:(k-1)}$. Then,

$G_{l(j)}$ in $S_{l(j)}$ = data in G_j whose $x_{v:(k-1)} \leq x_{(c),v:(k-1)}$, and

$G_{r(j)}$ in $S_{r(j)}$ = data in G_j whose $x_{v:(k-1)} > x_{(c),v:(k-1)}$. (EQN 3.2b)

The optimum splitting value (c*,v*) is selected after screening all possible cuts (c,v) of node j (c= 1 to size of subgroup G_j; and v = 1 to p). This systematic screening procedure is applied to each nonterminal node j, until the full sized tree model is obtained.

The sweeping operation in (EQN 3.1) is performed on variables $y_{:(k-1)}$ and $x_{v:(k-1)}$ (v ≠ k*) separately for subgroups G_{2j} and G_{2j+1}. If all the data points shown in Figure 3.1 represent subgroup G_j lying in subregion S_j. The piecewise linear function of $x_{k*:(k-1)}$ (solid line) represents the partial leverage residual plot for x_k on S_{2j} and S_{2j+1}.

3.3 "Outlier" Detections

Stratifying on $x_{v:(k-1)}$ can isolate high leverage data points indicated by extreme values of $x_{v:(k-1)}$. Let N_{min} denote the minimum group size to fit the linear functions, $L_{l(j)}$ or $L_{r(j)}$, in (EQN 3.2a). The first and last few cuts (c,v) of each variable $x_{v:(k-1)}$ (v = 1 to p) can be used to check the presence of "outliers" by fitting

$$L_{l(j)}(X_{k-1}) = L_j(X_{k-1}) + s_{l(j)}1, \quad L_{r(j)}(X_{k-1}) = L_j(X_{k-1}) + s_{r(j)}1 \qquad \text{(EQN 3.3)}$$

to $G_{l(j)}$ and $G_{r(j)}$ for $c \leq N_{min}$, and $c > N_j - N_{min}$ respectively.

3.4 Force-to-enter Variables

Within this framework, local models $L_j(X_{k-1})$ in (EQN 3.2a) can be extended to include q force-to-enter variables Z_q. Let $M_j(X_{k-1}, Z_q)$ and $L_j(X_{k-1})$ be the local models on subregion S_j. Then from

(EQN 3.2a), $M_{2j}(X_k,Z_q)$ and $M_{2j+1}(X_k,Z_q)$ can be derived from $L_j(X_{k-1})$ as follows,

$$M_{2j}(X_k,Z_q) = L_j(X_{k-1}) + s_{2j}1 + r_{2j}x_{k*:(k-1)} + Z_{:(k-1)}t_{2j}, \quad \text{(EQN 3.4)}$$

$$M_{2j+1}(X_k\ Z_q) = L_j(X_{k-1}) + s_{2j+1}1 + r_{2j+1}x_{k*:(k-1)} + Z_{:(k-1)}t_{2j+1}$$

In this model, stratifications are performed on residual variables $x_{v:(k-1)}$ (v = 1 to p-q) whose effects are to be removed, but not on variables $z_{u:(k-1)}$ (u = 1 to q) whose effects are to be estimated. $\hat{t}_{2j}$ represents the estimated effects of the force-to-enter variables Z_q in subgroup G_{2j}, linearly adjusted for local effects of the k-1 variables X_{k-1} selected from previous splits.

Using this formulation, a fixed set of coefficient parameters t for Z_q will be estimated for each subgroup obtained. For this model, Siu (1985) proposed an objective function to choose splits which commensurate the bias with the variance of the estimates $\hat{t}$.

This force-to-enter option can be particularly useful in some applications. Consider the problem of applying recursive partitioning regression to analyze prospective randomized studies: the resulting tree model would be difficult to interpret when stratification is also performed on treatmen variables.

3.5 Interpretations

The coefficient parameters in the hierarchy of piecewise linear model are interpretable, allowing graphical assessment of nonlinearity in the data. For example, one can plot the individual effect of z_u (u = 1 to q) against x_v (v= 1 to p) to assess interactions between these variables. The interaction plot shown in Figure 3.1 is obtained by plotting $\hat{t}_{i,u}$, the estimated coefficient parameter of z_u for individual i, versus $x_{i,v}$ (i = 1 to size of entire training sample, v = 1 to p).

Distributions of variables X_p in each subgroup provide information on the compositions of these optimal partitions.

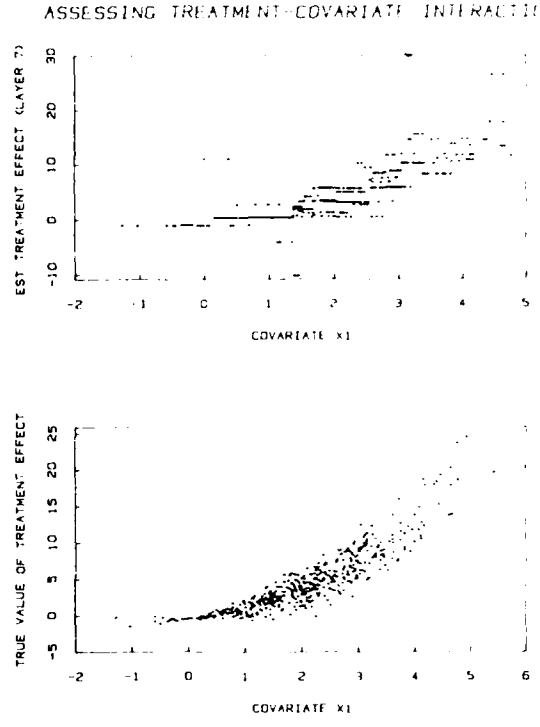


Fig. 3.1 True Model is $y = x_1 x_2 z + e$

3.6 Friedman's Model

Friedman (1979) presented an interesting approach to build tree-structured piecewise linear models. Specifically, a global multiple regression

$$L_1(X_{all}) = X_p a_p \quad \text{(EQN 3.5a)}$$

is fit to G_1 in S_1, the entire training sample. The effects of subsequent splits are to modify coefficients of these p variables one at a time. That is, in splitting node j, the coefficient of x_{v*} is modified by fitting

$$L_{2j}(X_p) = L_j(X_p) + s_{2j}1 + r_{2j}x_{v*}, \quad \text{(EQN 3.5b)}$$
$$L_{2j+1}(X_p) = L_j(X_p) + s_{2j+1}1 + r_{2j+1}x_{v*}$$

to

$$G_{2j} \text{ in } S_{2j} \text{ - data in } G_j \text{ whose } x_{v*} \le x_{(c*),k*}, \quad \text{(EQN 3.5c)}$$
$$G_{2j+1} \text{ in } S_{2j+1} \text{ - data points in } G_j \text{ whose } x_{v*} > x_{(c*),k*}$$

respectively.

As shown in (EQNs 3.2 and 3.5), the key differences between the two tree growing algorithms are choices of L_1 and the orthogonalization

process applied to the predictor variables **X**. These lead to some fundamental differences between the two methods: 1) the parent model L_j in this method is not nested within the child models L_{2j} or L_{2j+1}, 2) rectangular splits are used here to group data points, via stratification on the original (EQN 3.5b) instead of the residual (EQN 3.2b) variables, and 3) all of the predictor variables are used for stratification in this formulation.

The orthogonalization approach (EQN 3.2) is a natural extension of least squares method. It provides a simple framework to develop tree-structured methods for estimating iterative weighted least squares models. To build local models L_j in a forward stepwise manner has an added advantage. The score test in GLM provides the theoretical basis for the splitting rule developed in Section 4. From (EQN 3.5), such an extension does not seem to be feasible without major modifications of the fitting algorithm.

4. GENERALIZED REGRESSION TREE MODELS

4.1 Brief Review

A generalized linear model in Nelder and Wedderburn (1972) is defined by three components: the error structure given by one-parameter exponential family of the response variable y, the linear predictor $L(\mathbf{X}_p)$, and the link function $g(\mu_p)$ (i.e. $y = \mu_p + \varepsilon$, and $g(\mu_p) = L(\mathbf{X}_p) = r_0 1 + r_1 \mathbf{x}_1 + \ldots + r_p \mathbf{x}_p$).

For this model, Peduzzi et al. (1980) suggested to use the score test (Rao, 1973) for stepwise selection of variables. Full iteration is required only when the selected variable $\mathbf{x}_{k*}$ enters the model $L(\mathbf{X}_{k-1})$, but not for screening competing models. Pregibon (1982) showed that the score test in GLM can be computed using a normal linear model setup. In particular, the score statistic for H_o: $r_k = 0$ in $L(\mathbf{X}_k)$ is given by the additional regression sum of squares due to $\mathbf{x}_k$ in the weighted least squares regression of y^* on $\mathbf{X}_k$ (EQN 4.2). That is, for $\mathbf{e} \approx N(0, \mathbf{V}^{-1})$ and $\mathbf{V} = \hat{var}(\mathbf{y})$,

$$y^* = L(\mathbf{X}_{k-1}) + r_k \mathbf{x}_{k:(k-1)} + \mathbf{e} = L^1(\mathbf{X}_k) + \mathbf{e} \qquad \text{(EQN 4.2)}$$

with the weight and "working response variable" $y^* = \hat{L}(\mathbf{X}_{k-1}) + \mathbf{V}^{-1}(y - \hat{\mu}_{k-1})$ evaluated at the mle $\hat{r}_{k-1}$ from $L(\mathbf{X}_{k-1})$.

4.2 Tree-structured Extensions

This section discusses the use of optimal tree-structured approach for one-parameter exponential family models. The generalizations discussed here exploit the close connections between the stepwise approaches to fit least squares regression (EQN 3.1) and tree-structured normal response model (EQN 3.2a). The two procedures for adding variables to the models are identical except the recursive partitioning operations that lead to the fitting of regression tree models.

As shown by Nelder and Wedderburn (1972), the maximum likelihood estimates of GLM can be obtained through iterative weighted least squares method. Using the orthogonalization approach, the method discussed in Section 3 can be readily applied to build tree-structured exponential family models. Computation efforts can be saved by using an objective function analogous to the score test in GLM to choose optimal partitions of node j.

Following the formulations in Section 3.2, the cut at (c,v) is defined by replacing (EQN 4.2) with a piecewise weighted least squares regression of $y^*_{:(k-1)}$ on $\mathbf{x}_{k:(k-1)}$. That is, fitting

$$L^1_{l(j)}(\mathbf{X}_k) = L_j(\mathbf{X}_{k-1}) + s_{l(j)}1 + r_{l(j)}\mathbf{x}_{v:(k-1)}, \qquad \text{(EQN 4.3)}$$

and

$$L^1_{r(j)}(\mathbf{X}_k) = L_j(\mathbf{X}_{k-1}) + s_{r(j)}1 + r_{r(j)}\mathbf{x}_{v:(k-1)}$$

to $G_{l(j)}$ in $S_{l(j)}$ and $G_{r(j)}$ in $S_{r(j)}$ respectively. $G_{l(j)}$, $S_{l(j)}$, $G_{r(j)}$ and $S_{r(j)}$ are defined as in (EQN 3.2b). It is obvious that (EQN 3.2a) is a special case of (EQN 4.3), where $L^1_{l(j)}$ and $L^1_{r(j)}$ represent the one-step approximations to the mle of $L_{l(j)}$ and $L_{r(j)}$.

As in stepwise procedures for GLM, iterations will be performed to obtain the mle of the selected models. $L_{2j}(\mathbf{X}_k)$ on S_{2j} and $L_{2j+1}(\mathbf{X}_k)$ on S_{2j+1}, where $\mathbf{X}_k = [\mathbf{X}_{k-1}, \mathbf{x}_{k*}]$. The optimum split at (c*,k*) is defined by the piecewise linear model (L_{2j}, L_{2j+1}) which yields the largest

additional sum of squares due to $x_{k*:(k-1)}$ in (EQN 4.3). This method is identical to the usual stepwise procedure for GLM, if no splitting is performed.

4.3 "Outlier" Detections

Using this framework, potential outliers at extreme values of $x_{v:(k-1)}$ may be detected by fitting the weighted least squares regressions of

$$L^1_{l(j)} = L_{l(j)} + s_{l(j)}l \qquad \text{(EQN 4.5)}$$
$$L^1_{r(j)} = L_{r(j)} + s_{r(j)}l$$

to $G_{l(j)}$ when $c \leq N_{min}$, and to $G_{r(j)}$ when $c > N_j - N_{min}$.

4.4 Givens Rotations

The success of this computation intensive method depends on a reliable, efficient algorithm to compute and update the weighted least squares models. As the cut moves from (c,v) to (c,v+1,, one data point is added to $L_{l(j)}$ and then removed from $L_{r(j)}$. Givens rotations provide the basic computation method to update the piecewise linear model $(L_{l(j)}, L_{r(j)})$ in screening the candidate splits. Tne method is designed to identify extreme values and exclude them from further analysis. This may help improve the stability of the algorithm for deleting data points.

5. DISCUSSIONS AND CONCLUSIONS

This paper describes the basic methodology for using orthogonalization approach to fit a wide class of tree-structured models.

The versatility of this orthogonalization approach is illustrated by the large class of parametric and nonparametric hierarchical models that can be fit within this framework. A model specification rule is developed to clarify the property and uses of these procedures. It also helps explain differences among the tree-structured models in this class and their connections with the usual linear models.

In particular, procedures are specified by:

1) a split indicator (yes/no) for determining whether recursive partitioning is conducted or not;

3) an error structure (eg. normal, binary etc.);

4) types of predictor function (constant, linear); and

5) an objective function for choosing the optimal partition of each node (eg. score test, mean square error of the estimates $\hat{t}$ ... etc.).

To enhance flexibility of the procedure, three options are included. User can specify 1) force-to-enter variables, 2) prior weights, and 3) use of quantile splits. Prior weights can be used to obtain test sample error estimate. Issues which we do not have space to discuss here include practical problems of applying optimal stratifications to discrete response models, choices of objective function for choosing the "best" split of each node, design of effective output, descriptions of intermediate results, strategies of finding optimal-sized trees ... etc. For the normal response models, some of these points have been studied in Siu (1985). They provide the basis for developing the tree-structured extensions of generalized linear models in this paper.

REFERENCES

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984), Classification and Regression Trees, California: Wadsworth.

Friedman, J.H. (1979), "A tree-structured approach to nonparametric multiple regression," in Smoothing Techniques for Curve Estimation, eds. Gasser, Th. and Rosenblatt, M., Berlin: Springer-Verlag, pp. 5-22.

Mosteller, F., and Tukey, J.W. (1977), Data Analysis and Regression, Reading. Mass. Addison-Wesley.

Nelder, J.A., and Wedderburn, R.W.M. (1972), "Generalized linear models," Jour. Royal Statist Soc., Ser. A, 135, 370-384.

Pregibon, D. (1982), "Score Tests in GLIM with Applications," in GLIM 82: Proceedings of the International Conference on Generalized Linear Models, ed. Robert Gilchrist, New York: Springer-Verlag.

Siu, C.O. (1985), "Piecewise linear tree-structured regression with an application for the removal of confounding effects," unpublished Ph.D. dissertation, University of Toronto, Dept. of Statistics.

Siu, C.O., and Andrews, D.F. (1985), "Piecewise linear tree-structured regression with an application for covariance analysis," in Proceedings of the Statistical Computing Section, American Statistical Association, pp. 215-219.

Sonquist, J. A., and Morgan, J. N. (1964), The detection of interaction effects, Ann Arbor: Institute for Social Research, University of Michigan.

A STOCHASTIC EXTENSION OF PETRI NET GRAPH THEORY

Lisa Anneberg, Wayne State University

INTRODUCTION

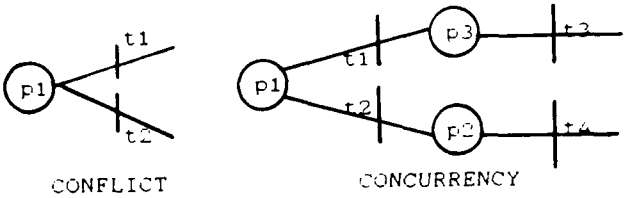
A tool of rising importance in the area of computer software analysis is the Petri Net. Petri Nets were developed in 1962 by Carl Adam Petri in West Germany. Since then, many applications and methods of analysis have been proposed by Petri and other authors. Petri Nets are used in modeling of a system, and then for the system's subsequent analysis. Petri Nets have some distinct advantages over graphical or other modeling and analysis techniques, most particularly the ability to depict concurrency and parallelism. Also, Petri Nets allow modeling at different levels of abstraction, further extending their usefulness.

It is proposed to extend Petri nets to include elements of stochastic behavior and to utilize these Petri nets for practical examples. Some elements of net theory [1] may not be applicable, but the contribution of improved graphical representation and reachability tree analysis is considerable. The stochastic behavior postulated answers "with what probability will the nodes in this path function?" (given the non-determinancy of firing rules, an essential element in Petri Net theory).

PETRI NET DEFINITIONS

Petri Nets have an outstanding advantage of the ability to show parallelism in concurrent systems, in addition to showing elements of control. This is an especially useful advantage when discussing computer hardware of LSI or greater complexity (as it is or should be highly parallel).

Petri Nets are a bipartite graph capable of modeling well a wide variety of situations. The two types of nodes are places (represented by circles) and transitions (represented by bars). The nodes are connected by (usually directed) arcs. Tokens are graph primitives that provide control. The tokens reside in the places and represent an item or condition (for example data or machines). The tokens move or flow when a transition 'fires'. A transition may fire when each input place contains at least one token in it. After firing, each outgoing place from that transition will contain an additional token. Generally, places may contain more than one token. Conflict and non-determinancy are allowed, and can be advantageous in modeling real systems. The following diagrams illustrate this conflict and concurrency:



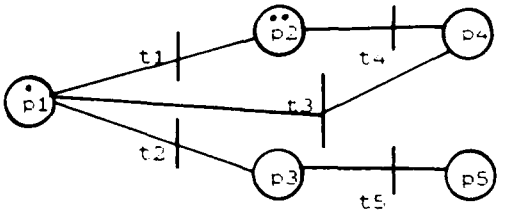
CONFLICT CONCURRENCY

In the CONFLICT diagram, t1 and t2 will not be able to fire simultaneously. The token from p1 can enable them both, but only one transition may fire. In the CONCURRENCY diagram, t3 and t4 can fire at the same time (in parallel) or in some other specified synchronous manner.

Formally, Petri Nets are represented by a four-tuple PN :

PN = (P, T, I, O) where
- P = (p1, p2,....,pn), the set of n given places
- T = (t1, t2,....,tm), the set of m given transitions
- I = (I(t1), I(t2),....,I(tm)), the set of input places to each transition
- O = (O(t1), O(t2),....,O(tm)), the set of output places to each transition

The marking Mi is a set expressing token number for every place at a time i. An example Petri Net is:



The marking is (1, 2, 0, 0, 0). P = (p1, p2, p3, p4, p5). T = (t1, t2, t3, t4, t5). I(t1) = p1, I(t2) = p1, I(t3) = p1, I(t4) = p2, I(t5) = p3, O(t1) = p2, O(t2) = p3, O(t3) = p4, O(t4) = p4, O(t5) = p5.

One popular method of Petri Net analysis is that of reachability analysis. Reachability analysis, first proposed by Murata [5], involved the creation of a p x t matrix. This matrix has n rows, where n is the number of places and m columns, where m is the number of transitions. This matrix illustrated the 'connections' between places and transitions. A zero entry would occur where the place and transition were not connected. In short, the p x t matrix is the incidence matrix, where places and transitions are connected.

Reachability analysis can be utilized to determine system success paths for reliability evaluation [4]. The state equation is formulated [5]:

$$A^T \Sigma = \Delta M$$

where A^T is the transpose of the incidence matrix (p x t), ΔM is the change in marking, and Σ is the firing count vector (1 when column is included in the path). An example of reachability analysis is presented in the next section.

STOCHASTIC BEHAVIOR

Simply stated, a stochastic process is one developing in time and governed by probabilistic behavior of some type. A Petri Net can be a stochastic process, if probabilities are associated with some relevant features. In the literature [1], probabilities have been associated with either transitions or places. It is proposed that both places and transitions can and should have associated probabilities. The inherent difficulties in the mathematics will not be greater, if both node types are considered to behave probabilistically. This association will not change the determinancy of the firing of the transition, but will state the probability of firing once 'chosen'.

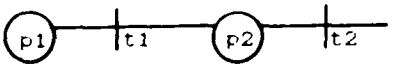
The given p x t matrix for analysis:

```
|P(p1)P(t1)  P(p1)P(t2)  ...  P(p1)P(tm)|
|P(p2)P(t1)  P(p2)P(t2)  ...  P(p2)P(tm)|
|P(p3)P(t1)  P(p3)P(t2)  ...  P(p3)P(tm)|
      .                          .
      .                          .
      .                          .
|P(pn)P(t1)  P(pn)P(t2)  ...  P(pn)P(tm)|
```

Where P(pi) is the probability of success associated with place i and P(tj) is the probability of success associated with transition j. For example, if place m and transition n are connected, and their respective probabilities are .9 and .8, the correct probability of that path operation is 0.72.

However, if the path is obtained through this reachability analysis and the reachability matrix is utilized to determine the total system probability, the non-terminal nodes (thoses in the interior, that is having both output and input nodes) will have their probabilities counted twice.

To illustrate, the following path:



will have associated reachability probabilities calculated as:

P(p1)P(t1)P(t1)P(p2)P(p2)P(t2)

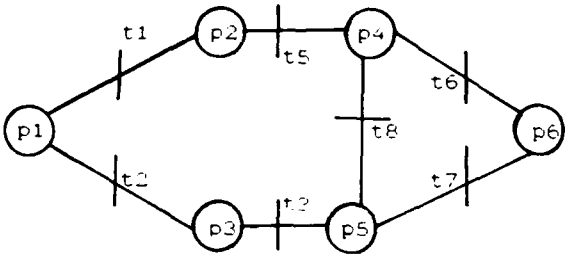
The interior nodes t1 and p2 will have the probabilities counted twice.

To counteract this effect, a small routine should be utilized in conjunction with the formulation of the matrix when calculating the path probabilities:

```
BEGIN
  Read i = 1 , m      :   places
    For i = O(tj) or I(ti),
        Delete  P(pm)  for second
        Incidence.
  Read j = 1 , n      :  transitions
    For j = O(pi) or I(p1),
        Delete P(tn) for second
        Incidence.
END
```

A routine such as this will counterect the affect of counting each interior node twice (it should be no more than a pair since self-loops are normally not allowed in Petri Nets for ease of calculation). As previously stated, Petri Nets are advantageous in that much of the analysis is easily performable on a computer, and this routine clearly is.

An example of this routine is given utilizing Petri Nets:



A^T and ΔM for this Petri net are:

	t1	t2	t3	t4	t5	t6	t7	t8
p1	1	1	0	0	0	0	0	0
p2	1	0	0	0	1	0	0	0
p3	0	1	1	1	0	0	0	0
p4	0	0	0	1	1	1	0	1
p5	0	0	1	0	0	0	1	1
p6	0	0	0	0	0	1	1	0

$$\Delta M = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

This methodology technique, after A^T and ΔM calculation, is column comparison. The comparison is accomplished via column addition. The formulation of this problem is as follows:

```
SET  L   =   column number
             (transitions number).
INITIAL  L = 1.
BEGIN    L = L + 1.
         Add all combinations of L
         columns.
         Compare to   M.
         If  L =   M.  success path
         exists.
         If pn = pm (self-loop),
            disallow success path
         RETURN to BEGIN.
```

Once the entire set (and this does require complete enumeration) is compared, the exhaustive set of paths through the Petri Net will have been obtained.

The success paths obtained for this Petri Net example are:

```
p1-t1-p2-t5-p4-t6-p6
p1-t2-p3-t4-p4-t6-p6
p1-t2-p3-t3-p5-t7-p6
p1-t2-p3-t4-p4-t8-p5-t7-p6
p1-t2-p3-t3-p5-t8-p4-t6-p6
```

In addition to calculating probabilites associated with correct function of these paths (assume given probabilities for each specific node), calculation of time required (given some associated time value for each specific node) can also be made. To calculate times necessary, times for each node will be added (same allowance for interior node must be made).

CONCLUSION

A methodology to calculate stochastic probabilities in a self-loop free Petri Net is presented. It is based on reachability matrix and analysis. This method will calculate success path probabilities for Petri Net success paths. A further extension will allow time calculation for the success paths of the Petri Net to be made.

A promising area for further research would be to develop a method or heuristic to reduce the total combinations reviewed. Further, study in the Markov behavior of given Petri Nets should be made.

A major difficulty [7] is that the more elaborate the Petri Net (the better it models the 'real-world') the more complex the calculations become. Associating time, cost, or probabilities with the Petri Net may make calculations complex or theoretical formulations non comlplete.

ACKNOWLEDGMENT: I want to extend my thanks to Dr. R. L. Thomas and the Institute of Manufacturing Research, in Detroit, Michigan.

REFERENCES

1. W. Brauer, W. Reisag, and G. Rosenberg. **Petri Nets: Central Models and Their Properties**. Springer-Verlag Inc., New York, N.Y., 1987.

2. G. S. Hura, 'Enumeration of All 2-Trees in a Graph Through Petri Nets', **Microelectronics and Reliability**, Vol. 23, Pages 851-853, 1983.

3. G. S. Hura, 'Enumeration of All Simple Paths in a Directed Graph Using Petri Net: A Systematic Approach', **Microelectronics and Reliability**, Vol. 23, Pages 157-159, 1983.

4. G. S. Hura, 'Simplification of Boolean Functions Through Petri Nets', **Microelectronics and Reliability**, Vol. 23, Pages 471-475, 1983.

5. T. Murata, 'Circuit Theoretic Analysis and Synthesis of Marked Graphs', **IEEE Transactions on Circuits and Systems**, Vol. CAS-24, No. 7, 1977, Pages 400-405.

6. E. Parsen, **Stochastic Processes**. Holden-Day, Inc., San Francisco, CA 1962.

7. J. Petersen, **Petri Net Theory and the Modeling of Systems**, Prentice Hall, Inc., Englewood Cliffs, NJ, 1981.

8. A. N Shirvayev, **Graduate Texts in Mathematics: Probability**, Springer-Verlag, Inc., New York, NY, 1984.

TIMED NEURAL PETRI NET

Nazih Chamas* and Harpreet Singh*

*Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202

ABSTRACT

The concept of a timed neural Petri Net is presented which is to be isomorphic to neural net architecture. The standard technique of neural net architecture is applied to this net. This basic Petri Net concepts have been extended. Some examples are presented.

INTRODUCTION

The Petri Net concept has proven to be a very powerful tool in modeling parallel, sequential and concurrent processing [5,7]. Even though standard Petri Net has a wide range of capabilities for modeling systems, many authors found it somewhat restrictive for general systems. Several extentions were proposed. In this paper we propose another extention which is aimed at expanding the modeling capabilities of Petri Net and reducing the complexity in parallelism. Most of what has been written about the similarities and differences between brain and machine [3,4,5] have explored a new era in computer designs including massive parallelism and functional modularity.

In the first section, the basic definitions of the human brain and neuron are presented. Section II deals with the basic definitions of Petri Net. In section III a review of some extensions to standard Petri Net has been presented. While in section IV a new mathematical formulation has been defined. In V a new reachability concept has been prooven. In VI the TNPN has been proposed. In VII an example has been given and finally some conclusions are drawn in section VIII.

I.1 Neuron

A neuron is the basic functional unit in the brain. It is small in area and volume but it is long. It is able to receive hundreds of inputs through the dentrites and process them in the cell body (soma). The result of process is the neuron output. The output goes through the axon hillock, axon and axon terminals to other neurons. Figure 1 is a typical neuron.

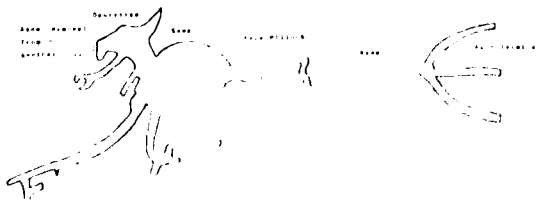


FIG. 1 A typical neuron

A neuron consists of synapses, soma, dentrites, axon, axon hillock and axon terminals.

*Soma: Soma or cell body is the neural process place. This place receives its input from other neurons through the dentrites and transmits the output through the axon.

*Dentrites: Dentrites are small, thin branches around the cell body. These dentrites receive the input data from the synapses and carry them to the soma.

Synapses: Synapses are the connection between the dentrites of one neuron and the axon terminals of an other neuron.

Axon hillock: Axon hillock works as a threshold unit. The output data cannot flow through the axon until its weight is equal to or exceeds the weight of the axon hillock.

Axon: There is one axon in each neuron. An axon is a thin, long channel. This channel terminate by a branch of terminals. These terminals are inputs to other neuron.

The neuron is a powerful cell. It is able to receive thousands of inputs and transmit thousands of outputs concurrently. The capability of this small cell and the archict within the neural net are good areas in research. For many years, researchers have been working very hard to make machine works similar to the human brain.

I.2 Human Brain:

The human brain is one of the most complex structure nets in the known universe. Over 100 billion individual neurons are grouped together to create a system with many separate areas (modules). Each area is able to process a specific type of data [3,4]. The cells in each area are grouped together to create a subsystem of hierarchical superposition, permitting information to flow in a stratified manner, layer by layer [4]. Each layer is defined as a level. Each level is able to process data at a certain degree of complexity. The lower layer has the higher capabilities in processing. Cellular interconnections between and within layers are well organized. The organization is comprised of parallel and hierarchical architectures. Parallelism is between layers, while the hierarchical architecture is within each layer. Computers are much faster in speed, but they perform poorly in tasks that emulate the natural information processing that humans handle routinely. The amazing brain architecture is the secret of brain capabilities. Researchers have very good results in integrated circuits. Thousands of gates can be on a single chip less than inch in diameter. The new technology will reduce the complexity to build a powerful intelligence machine.

II. Basic Definition of Petri Net (PN)

Petri Net is a formal directed graph for analysing [illegible] systems and concurrent systems [5]. The first bipartite represented by circles is called place. The second usually represented by bars is called transitions. Arcs connect transitions to places and vice versa. If an arc exists from a place to a transition, from a transition to a place, then the place is called an input (output) to the transition. Therefore, the set of arcs is partitioned into input arcs (A) and output arcs (A). See figure 2.

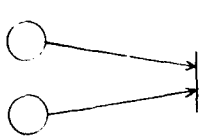


Fig. 2a. A place is an input to a transition

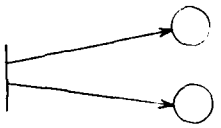


Fig. 2b. A place is an output of a transition

Formally, a Petri Net structure consists of three tuples.

$$PN = (P, T, A) \qquad (1)$$

Where $P = \{p_1, p_2, \ldots, p_n\}$ is the set of places

$T = \{t_1, t_2, \ldots, t_m\}$ is the set of transitions

and $A_i \subseteq (P \times T)$ is the set of transition input arcs

While $A_o \subseteq (T \times P)$ is the set of transition output arcs

and $A = A_i \cup A_o$

The places may contain tokens represented by dots. When tokens are present the Petri Net is called marked Petri Net. The execution of a marked Petri Net is governed by the following: A transition is enabled when all of its direct input places contain at least one token. An enabled transition can fire, this removing one token from each input place and placing one token in each output place. After a transition fires a new distribution of tokens occurs, thus, producing a new marketing.

In a more formal way, we can say that the marked PN is defined as

$$PN = (P, T, A, M_0) \qquad (2)$$

where P, T and A are as in (1) and $M_0 = m_{01}, m_{02}, \ldots, m_{0n}$. Where M denotes the distribution of tokens in the places in the initial marketing.

III. Some Extensions to Standard Petri Nets

Many extensions were introduced [1, 2, 3] and many other extensions will be introduced to the standard Petri Net increase the modeling power of the tool. The extensions considered in this section, are the definition of timed Petri Net (TPN), multiple arcs, inhibitor arcs, Stochastic Petri Nets (SPN). Timing usually plays a fundamental role in the description of the behavior of any system. The standard PN is able to describe only the logical structure of system and not their time evaluation. There are different ways to introduce time into a standard PN model, which allows the description of the dynamic behavior of systems and takes into account both the state evolution and the duration of each action performed by the system [5, 17]. A *formal definition of a TPN is thus as following:*

$$TPN = (P, T, A, M_0, Q) \qquad (3)$$

Where P, T, and A are as in (2) and $Q = Q_1, Q_2, \ldots, Q_n$ is the set of delays associated with PN

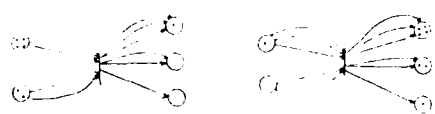
Stochastic Petri Net (SPN) models are obtained by associating with each transition in a PN as a exponentially distributed random variable that expresses the delay from the enabling to the firing of the transition. Formally, we can say that a SPN is defined as

$$SPN = (P, T, A, M_0, L) \qquad (4)$$

Where P, T, A, and M are as in (3) and $L = l_1, l_2, \ldots, l_m$. L is the set of the possible marking-dependent firing rates associated with the PN transitions.

Many other extensions were introduced to the standard PN to increase the modeling power of the tool. Some of these extensions are the multiple arc, inhibitor arc and the modeling of parbegin, and parend in PN, which will be defined by small examples.

In the multiple arc, as shown in Figure 3, more than one arc is allowed to connect a place to a transition and a transition to a place. A transition is enabled when all its direct input places contain at least the number of tokens equal to the number of arcs between the place and the transition. An enabled transition can fire, thus removing the number of tokens of each place equal to the number of arcs between the place and the transition and putting the arcs between the transition and the place



A PN with multiple arcs

A compact representation of multiple arcs

Fig. 3.

IV. Mathematical Formulation of the conflict equation

Consider the Petri Net in Figure 4 which we will decompose into different levels. The first level is formed by all the transmitting places.

$$P_1^1, P_2^1, \ldots, P_{n1}^1$$

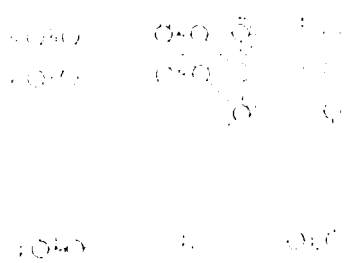


Fig. 4. Decomposition of a Petri Net into levels and classes

The information from any one of these places can flow to a connected place belonging to the second level through an appropriate transition

$$P_1^2, P_2^2, \ldots, P_{n2}^2$$

and so on. The places of a typical level i are denoted by

$$P_1^i, P_2^i, \ldots, P_{ni}^i$$

This level can receive direct information from places of the i-1 level and transmit the information directly to places of the i+1 level. The last level labeled by L is formed by all the receiving places. It is now obvious that the total number of places is

$n_1 = n_2 = \ldots = n_L$

The second level will be divided into n clases

$$C_1^2, C_2^2, \ldots, C_{n1}^2$$

where $C_1^2 = \{k \mid P_1^1$ is connected to $P_k^2\}$

In general the ith level where $2 < i < L$ is divided into n classes.

$$C_1^i, C_2^i, \ldots, C_{n-1}^i$$

where: $C_1^i = \{k \mid P_1^{i-1}$ is connected to $P_k^i\}$

Note that C_l^i where $1 < l < n_{i-1}$ are not necessarily disjoint however

$$\bigcap_{l=1}^{n_{i-1}} C_l^i = 1, 2, \ldots, n_i$$

With these preliminary remarks the conflict equations between level i-1 and level i can be written as follows:

$$P_l^{i-1} \rightarrow P_k^i \text{ where } k \in C_l^i \quad (5)$$

In general the conflict equations between level i-1 and level j can be written as

$$P_l^{i-1} \rightarrow P_k^j \text{ where } k \in C_l^{j-1} \quad (6)$$

Equations (2) and (5) provide a step-by-step procedure from which the conflict equation between any two levels can be written

Now different representations of the conflict equation can be given. In the following we will focus on linear algebra representation using boolean arithmetic. We introduce a transition matrix

$$M^{i-1,i}$$

with entries

$$a_{kl} = \begin{cases} 1 \text{ if } k \in C_l^i \\ 0 \text{ otherwise} \end{cases}$$

If the information at level i-1 is represented by a column matrix A with entries X such that:

$$X_j = \begin{cases} 1 \text{ if } p_j \text{ is loaded} \\ 0 \text{ otherwise} \end{cases}$$

Then after the firing we have:

$$A^i = M^{i-1,i} \cdot A^{i-1} \quad (7)$$

Now the transition matrix between level i and L where $i < L$ is given by:

$$M^{i,L} = M^{i,i+1} M^{i+1,i+2} \ldots M^{L-1,L} \quad (8)$$

V. Reachability in the weak Sense

Now we introduce a new concept closely related to that of reachability [2,5] which we shall call reachability in the weak sense

Consider two places: P_l^i P_k^L

belonging to levels i and L, respectively. To know whether a token originally in P_l^i can reach P_k^L, we have to form the matrix $M^{i,L}$. If the entry a_{kl} is equal to 1 then the token can be transmitted to P_k^L otherwise $a_{kl} = 0$ it cannot. Knowing the reachability in the weak sense for all places one can immediately have information about the reachability for the net in large for the model we are considering.

VI. Timed Neural Petri Net (TNPN)

The structure type and behavior of Petri Nets almost correspond to the functioning of the fundamental neural elements. The Timed Petri Net has been modified to a Timed Neural Petri Net definition to increase the power of the tool and to accomodate almost all the elements of the neural system. The modifications are in the concepts of the place and the transition. Figure 6 is a typical cell in TNPN

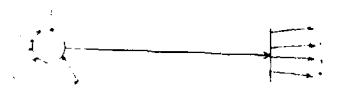


Fig. 6. A typical TNPN cell

This cell has only one place and one transition. The place has n inputs and only one output. The transition has m outputs and only one input. The place contains colored tokens. Each color has a different weight. The higher weight has the higher priority in processing and vice versa. The output arc works as a threshold [4]. Tokens are not allowed to flow to the transition from the place until their weight exceeds or equals the arc weight. The transition is able to process and transmit the transferable token to many places. The refinement transition is a subnet, see Figure 7 [9]. This subnet consists of input transition t and one output transition t'. The input transition t receives only one copy at a time and transmits k copies to k places concurrently. The output transition is an OR transition. The t' is enabled when any one of its input arcs is active. Then, t' is able to transmit copies to n places and so on.

The k places are input to k statements $s_1, s_2, \ldots, s_k$. The number of statements in each TNPN transition is equal to the number of colors which are used for the colored tokens. Each statement is designed to process specific colored tokens. For instance, when the k statements receive the k messages only one statement is going to process the message. The rest of copies in the (k-1) statements will disappear after the life time [3].

Finally, the similarities between the neuron and the TNPN are as follows:

- The place P_i represents the cell body (soma). It is able to receive many inputs and transmits the outputs through one arc.
- The arc between the place and the transition in TNPN represents the axon and it is working as a threshold [4].
- The arcs from the t_i to the places $P_1, P_2, \ldots, P_3$ are similar to the axon terminals.
- The colored tokens represent the chemicals in the neuron.
- The input arcs to the place P represent the axon terminals from another neuron.

Fig. 7 A refined TNPN transition

Figure 3 is basic Petri Net. The reachability from a place to another place or places is weak because of the conflict properties. But if we redesign the same net in Figure 3 with TNPN, we will get a new net. See Figure 5.

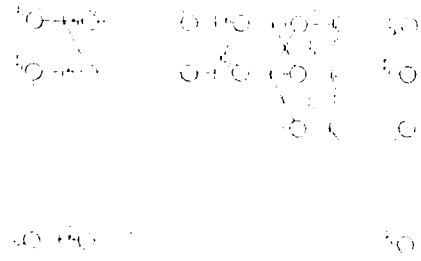


Fig. 5 Decomposition of a TNPN into level and classes

The main difference between the nets in Figure 3 and Figure 5 is the output of place P_1^1 in Figure 3 is able to flow to only one place in class C_1^{i+1} in level L_{i+1}. While the output of the place P_1^1 in Figure 5 flows to all places in class C^{i+1}. This property protects the net from the conflict complexity and make the reachability of weak sense a reachability of strong sense. So the net in Figure 3 is in conflict while the net in Figure 5 works in parallel.

VII. Example

The TNPN model in Figure 9 is based on a neural structure. This model is used to control the contraction of some muscles. In this Figure, the main neurons, which provide inputs to the muscle fibers are represented by places P_1^4 and P_{1-1}^4 in level L_4. Places in L_1 are represented as the pain receptor. L_2 are represented as fiber neurons, which tranfer the messages from L_1 to L_3. Places in L_3 level represent the human brain. The places in this level are able to process the input data from L_2 or L_4, and according to process results L_3 will send messages to L_5 through L_4. L_5 works mechanically. The degree of contraction of the muscle is proportional to the color of tokens at place P_1^4 and P_{1-1}^4 in L_4. After the contraction, P_{15}^5 in L_5 will send a message back to P_{13}^3 in L_3 through P_{14}^4 in L_4, indicating that the contraction is done. The places in L_3 are able to interrupt the contraction in L_5 by sending higher weight tokens to L_5 through L_4.

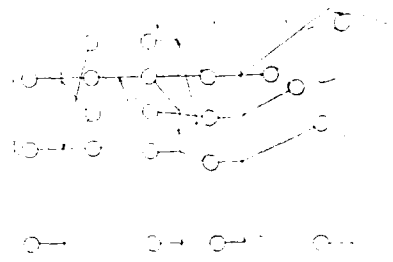


Fig. 9 A TNPN for neurons of some muscles

VIII. Conclusion:

The modeling of a TNPN cell has been discussed through meaningful interpretation of various Petri Net structure techniques. A mathematical interpretation has been given to a conflict equation inherently modeled by PN. A new concept has been introduced to reduce the conflict complexity by sending a number of messages equal to the number of outputs. This model can be considered as a basis for translating brain features into a framework of a computing system. The success in implementing this model will lead to a revolution in computer architectures and artificial intelligence systems.

References:

1. N. Chamas and H. Singh, "Petri Net Approach to a Modeling system Knowledgebase", 30th Midwest Symposium on Circuits and Systems, Syracuse Univ. Aug. 16-18, 1987.

2. A. A. Khan, H. Singh, "Petri Net Approach to Design and Development of Modern Computer Systems", Ph.D. Thesis, Electronics and Communication Engineering, Univ. of Roorkee, Roorkee-247672 (India) Sep. 1991.

3. Rosenblat, F., "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms", Spartan books, Washington D. C. (1961).

4. J. Metzler, "Systems Neuroscience", AP INC, 111 Fifth Ave., NY 10003 1977.

5. J. L. Peterson, "Petri Net Theory and Modeling of Systems", Prentice Hall Inc., Englewood Cliffs, N.J. 07632 1981.

6. C. Zervos, "Colored Petri Nets: Their Properties and Applications", Technical Report 107, System Eng. Lab., Univ. of MI., Ann Arbor, MI., Jan 1977.

7. T. Murata, "Petri Nets, Marked Graphs, and Circuit-System Theory", IEEE Transactions on Circuits and Systems, Vol. CAS-24, Number 7, (July 1977), pages, 400-405.

8. J. Meldman, "A New Technique for Modeling the Behavior of Man-Machine Information Systems", Sloan Management Review, Vol. 15, Number 3, Spring 1977, pp. 29-46.

9. J. Noe, "Abstraction and Refinement with Modified Petri Net", AFCET Journees sur les Reseaux de Petri (AFCET Workshop on Petri Nets), Paris, France, March 1977, pp. 157-160.

XIII. SIMULATION

ESTIMATING STANDARD ERRORS: EMPIRICAL BEHAVIOR OF ASYMPTOTIC MSE-OPTIMAL BATCH SIZES

Wheyming Tina Song and Bruce Schmeiser, Purdue University

Abstract

When an estimator of the variance of the sample mean is parameterized by batch size, one approach for selecting batch size is to pursue the minimal mean squared error. Recently asymptotic results have been obtained for the mse-optimal batch size. Based on Monte Carlo experiments we conclude that the asymptotic formula is an accurate approximation when used with finite-size samples from processes having geometrically decreasing autocorrelations, even when the ratio of sample size to the sum of autocorrelations is as small as five. The study considers three steady-state data processes, four estimator types, and four sample sizes. Although we don't discuss batch-size estimation procedures, the formula is a foundation for estimating the optimal batch size from data in practice.

1. Introduction

Estimating the variance of the sample mean is a fundamental problem of statistics. In simulation experiments and some other contexts, the data are sometimes assumed to be from a covariance-stationary process X having unknown mean μ, unknown positive variance R_0, and unknown finite fourth moment μ_4. Various types of estimators have been proposed, including regenerative [2], ARMA time-series models [3,4,12], spectral [8], standardized time series [7,13], nonoverlapping batch means [11], and overlapping batch means [10]. The batch-means and some standardized-time-series estimators operate on batches of observations; therefore the statistical properties of such estimators depend on the batch size, m, as well as the process X and the sample size n.

We study here the mean squared error (mse) of nonoverlapping-batch-means (NBM), overlapping-batch-means (OBM), standardized-time-series-area (STS.A), and the nonoverlapping-batch-means combined with standardized-time-series-area (NBM+STS.A) estimators. We apply these four estimator types to three data processes, each have geometrically decreasing correlation structure, with Bernoulli, normal, and exponential marginal distributions. Four sample sizes are considered.

Although we report mse's of various combinations of estimator, data process, and sample size, the focus here is on the accuracy of an asymptotic formula for batch sizes that minimize mse. In Section 2 we state the result in discrete time from Schmeiser and Song [14]. The result originally appeared in Goldsman and Meketon [6], but in continuous time and without the explicit constant γ_1.

2. Summary from Schmeiser and Song

We summarize here the asymptotic results in Schmeiser and Song [14] for comparison to the empirical results in Section 3.

For $h = 0, 1, 2, \cdots$, let ρ_h be the lag-h correlation $\mathrm{corr}(X_i, X_{i+h})$. Define the constants

$$\gamma_0 = \sum_{h=-\infty}^{\infty} \rho_h = 1 + 2 \sum_{h=1}^{\infty} \rho_h ,$$

and

$$\gamma_1 = \sum_{h=-\infty}^{\infty} |h| \rho_h = 2 \sum_{h=1}^{\infty} h \rho_h$$

For example, independent and identically distributed (iid) processes correspond to $\gamma_0 = 1$ and $\gamma_1 = 0$. These constants play a central role in determining asymptotic mse, as shown in Proposition 1.

Let m denote the batch size, $\hat{V}(m)$ denote the estimator of the variance of the sample mean, and $\mathrm{bias}(\hat{V}(m)) \equiv \mathrm{E}(\hat{V}(m)) - \mathrm{var}(\bar{X})$ denote the bias. For NBM, OBM, STS.A, and NBM+STS.A estimators, Proposition 1 holds.

Proposition 1. If both γ_1 and the fourth moment exist and are finite, then there are constants c_b and c_v such that

$$\lim_{m \to \infty} \lim_{n/m \to \infty} n\, m\, \mathrm{bias}(\hat{V}(m)) = -c_b \gamma_1 R_0 , \quad (2.1)$$

$$\lim_{m \to \infty} \lim_{n/m \to \infty} \frac{n^3}{m} \mathrm{var}(\hat{V}(m)) = c_v (\gamma_0 R_0)^2 , \quad (2.2)$$

and

$$\mathrm{mse}(\hat{V}(m)) \simeq \left(\frac{c_b^2 \gamma_1^2}{n^2 m^2} + \frac{m\, c_v\, \gamma_0^2}{n^3} \right) R_0^2 . \quad (2.3)$$

The optimal batch size, m^*, satisfies

$$\lim_{n \to \infty} n^{-1/3} m^* = \left[2 \left(\frac{c_b^2}{c_v} \right) \left(\frac{\gamma_1}{\gamma_0} \right)^2 \right]^{1/3} \quad (2.4)$$

and the optimal mse satisfies

$$\lim_{n \to \infty} n^{8/3} \mathrm{mse}(\hat{V}(m^*)) = R_0^2 \left(\frac{3}{2^{2/3}} \right) \left(c_b^2\, c_v^2\, \gamma_0^4\, \gamma_1^2 \right)^{1/3} \quad (2.5)$$

In terms of the correlation structure, the asymptotic bias is a function of only γ_1 and the asymptotic variance is a function of only γ_0. The asymptotic mse, optimal batch size, and optimal mse are all functions of both γ_0 and γ_1.

Goldsman and Meketon provide the constants c_b and c_v so that we can compare NBM, OBM, STS.A, and NBM+STS.A estimators in terms of their asymptotic mse's. Table 1, from Schmeiser and Song [14], reviews and extends their results.

Table 1: Comparison of NBM, OBM, STS.A, and NBM+STS.A Estimators

$\hat{V}$	NBM	OBM	STS.A	NBM+STS.A
c_b	1	1	3	2
c_v	2	4/3	2	1
$\lim_{n\to\infty} [2n(\frac{\gamma_1}{\gamma_0})^2]^{-1/3} m^* = (\frac{c_b^2}{c_v})^{1/3}$	0.79	0.91	1.65	1.59
$\lim_{n\to\infty} \frac{2^{2/3}}{3} n^{8/3} [(\gamma_0^4\gamma_1^2)^{1/3} R_0^2]^{-1} \text{mse}(m^*) = (c_b^2 c_v^2)^{1/3}$	1.59	1.21	3.30	1.59
$\lim_{n\to\infty} [3(\frac{\gamma_0^8}{2\gamma_1^2})^{1/3} R_0^2\, n^{-10/3}]^{-1} \frac{\partial^2 \text{mse}}{\partial m^2}\mid_{m=m^*} = (\frac{c_v^4}{c_b^2})^{1/3}$	2.52	1.47	1.21	0.63

The batch means methods have relatively little asymptotic bias; NBM and STS.A have relatively large asymptotic variance.

The optimal batch size constants are shown in the third row. The batch means estimators require batches about half the size of STS.A estimators.

The optimal mse constants are shown in the next-to-last row. OBM is smallest, with NBM and NBM+STS.A a little larger, and STS.A about double.

The last row contains a measure of the mse-robustness to batch size. Since a practitioner needs to estimate the optimal batch size, and since the statistical estimate will not always be correct, an appealing property of an estimator is that it not be sensitive to batch size in the region of m^*. We use the second derivative of the mse with respect to batch size as a measure of estimator robustness. This derivative, based on equation (2.3) and evaluated at any m^* satisfying equation (2.4), is

$$\frac{\partial^2 \text{mse}(\hat{V}(m))}{\partial m^2}\mid_{m=m^*} \simeq (\frac{c_v^4}{c_b^2})^{1/3} \frac{[3(\frac{\gamma_0^8}{2\gamma_1^2})^{1/3} R_0^2]}{n^{10/3}} .$$

The constant $(\frac{c_v^4}{c_b^2})^{1/3}$ is shown in the last row of Table 1. NBM+STS.A is the most robust and NBM is the least robust of the four estimator types.

3. Monte Carlo Results

In this section, we report some Monte Carlo experiments for finite sample sizes and compare the finite-sample results to the asymptotic results of Section 2. In Section 3.1 we discuss the three processes used in the experiments described in Sections 3.2 and 3.3. These three processes have identical correlation structure $\rho_h = \rho^h$, but different marginal distributions.

In particular, we compare the (finite-sample) optimal batch sizes m^* to the asymptotic optimal batch size

$$\bar{m}^* = 1 + \left[2n \left(\frac{c_b^2}{c_v} \right) \left(\frac{\gamma_1}{\gamma_0} \right)^2 \right]^{1/3} , \qquad (3.1)$$

which is motivated by equation (2.4) and the fact that $m^* = 1$ for iid data.

We also make three comparisons among mse's: the (finite-sample) optimal mse, denoted by $\text{mse}(m^*)$; the (finite-sample) mse with asymptotic batch size $\bar{m}^*$, denoted by $\text{mse}(\bar{m}^*)$; and the asymptotic mse evaluated at the asymptotic optimal batch size $\bar{m}^*$, denoted by $\bar{\text{mse}}(\bar{m}^*)$ and defined by

$$\bar{\text{mse}}(\bar{m}^*) = n^{-8/3} R_0^2 \left(\frac{3}{2^{2/3}} \right) \left(c_b^2 c_v^2 \gamma_0^4 \gamma_1^2 \right)^{1/3} . \qquad (3.2)$$

3.1 The Three Processes

The Monte Carlo experiment described in Section 3.2 is designed to investigate the effect of sample size and marginal distribution on the accuracy of $\bar{m}^*$ as an approximation for m^* when used with finite sample sizes. The marginal distributions are exponential, normal, and symmetric Bernoulli.

All three processes have the correlation structure $\rho_h = \rho^h$. For such a correlation structure, $\gamma_1 = (\gamma_0^2 - 1)/2$, so the optimal batch size for the four types of estimators we consider is a function of γ_0 and n only. Therefore, the conclusions may not be true for other correlation structures.

In the experiments, we want to specify the mean, μ (irrelevant); variance of the sample mean, $\text{var}(\bar{X})$,

the sum of correlations, γ_0, and sample size, n. So for each of the three processes, we give the structural definition and formulas for calculating parameter values from μ, var$(\bar{X})$, and γ_0. For simplicity, we use $\rho \equiv \rho_1$. For all three processes, $\rho = (\gamma_0 - 1)/(\gamma_0 + 1)$ and

$$R_0 = \frac{n\,\mathrm{var}(\bar{X})}{(\frac{1+\rho}{1-\rho}) - \frac{2\rho(1-\rho^n)}{n(1-\rho)^2}} .$$

EAR(1) process [9]:

$$X_t = \begin{cases} (\mu - \frac{1}{\lambda}) + \rho[X_{t-1} - (\mu - \frac{1}{\lambda})] & \text{w.p. } \rho \\ (\mu - \frac{1}{\lambda}) + \rho[X_{t-1} - (\mu - \frac{1}{\lambda})] + \epsilon_t & \text{w.p. } 1-\rho, \end{cases}$$

where ϵ_t is iid exponential with rate $\lambda = R_0^{-1/2}$.

AR(1) process [5]:

$$X_t = \mu + \rho(X_{t-1} - \mu) + \epsilon_t \ ,$$

where ϵ_t is iid normal with mean zero and variance $(1-\rho^2)R_0$.

The Symmetric Two-State Markov Chain (S2MC) [1]:

Let $\{X_i\}_{i=1}^n$ be a two-state dependent symmetric Bernoulli process with state space $\{c, d\}$ and transition matrix

$$\mathbf{P} = \begin{bmatrix} (1+\rho)/2 & (1-\rho)/2 \\ (1-\rho)/2 & (1+\rho)/2 \end{bmatrix},$$

where $c = \mu - R_0^{1/2}$ and $d = \mu + R_0^{1/2}$.

The S2MC result and the relationships among γ_0, R_0, and var$(\bar{X})$ for these processes are derived in Appendix A.

3.2 Experiment 1: Monte Carlo Results for the Three Processes

This section contains the optimal mse and optimal batch-size results of a Monte Carlo experiment using the three processes of the last section with lag-one correlation $\rho = 0.8182$. The estimator type is OBM. Sample sizes are 50, 500, 1000, and 5000. In all cases, var$(\bar{X}) = 1$; therefore the variance of the observation R_0 is a function of n. The mean is (arbitrarily) $\mu = 0$.

The results are based on 10000 independent observations at each design point. The mse's have standard errors smaller than .004. The optimal batch-size results reported are correct to within about one unit of the least-significant digit.

Table 2 shows a comparison of finite-sample optimal batch sizes and the asymptotic optimal batch size $\vec{m}^*$. The rows correspond to sample sizes and the columns correspond to process types. The right-most column shows the asymptotic batch size $\vec{m}^*$ as a function of n. Other entries are estimated optimal batch sizes based on the Monte Carlo experiment.

Table 2: A Comparison of Finite-Sample and Asymptotic Optimal Batch Size
Estimator Type : OBM
$\rho_h = (0.8182)^h$

n	EAR(1)	AR(1)	S2MC	$\vec{m}^*$
50	7	10	12	13
500	22	25	27	27
1000	29-30	31-32	34-35	34
5000	56-57	57	57-61	58

More than one batch size is shown in the last two rows ($n = 1000$ and 5000) because the mse function becomes flatter with increasing n, making Monte Carlo identification of the single best batch size difficult.

The asymptotic optimal batch sizes for all three processes should have the same value because of the common correlation structure. Indeed, when $n = 5000$ the Monte Carlo estimates are all essentially the same and equal to $\vec{m}^*$.

S2MC converges to $\vec{m}^*$ quickest, then AR(1), and finally EAR(1). Since EAR(1), AR(1), and S2MC have different marginal distributions (exponential, normal and symmetric Bernoulli), the values of the kurtosis (9, 3 and 1) also differ. We know that for finite sample sizes the optimal batch size depends upon the kurtosis; based on these Monte Carlo results we conjecture that the larger the kurtosis the slower the convergence.

A suboptimal batch size, even if distant from m^*, can be quite satisfactory if the associated mse is close to mse(m^*). Therefore, we now compare the differences between the optimal mse and the mse associated with the (possibly) suboptimal batch size $\vec{m}^*$.

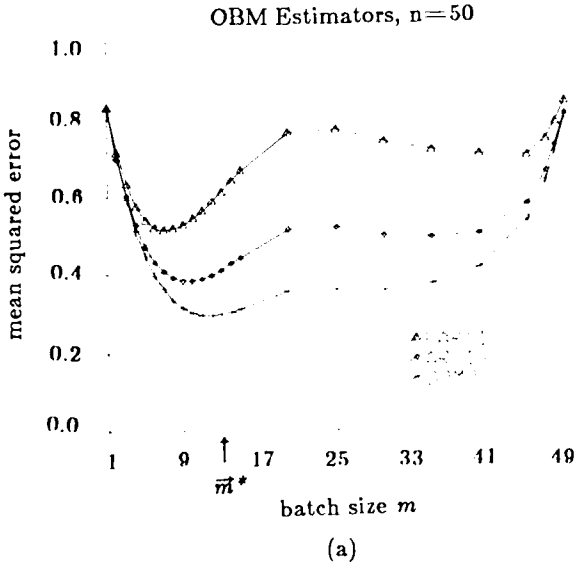
Figure 1 shows the (estimated) mse's for the three processes for sample sizes $n = 50$ and $n = 500$. In Figure 1 (and in Table 2) the largest difference between the optimal mse, mse(m^*), and the mse at the approximated optimal batch size, mse$(\vec{m}^*)$, occurs for $n = 50$ for the EAR(1) process, where the mse at $\vec{m}^* = 13$ is about 20% larger than the optimal mse at $m^* = 7$. In the other case shown, the difference between mse(m^*) and mse$(\vec{m}^*)$ is negligible.

So, at least for this correlation structure, the asymptotic batch size formula (3.1) accurately indicates a batch size having near-optimal mse for sample sizes as small as $n = 50$.

3.3 Experiment 2: Monte Carlo Results for Four Estimators

In this section, we investigate the accuracy of the asymptotic optimal batch size, $\vec{m}^*$, in estimating the optimal batch size for four types of estimators: NBM, OBM, STS.A, and NBM+STS.A. The data process is

AR(1) with $\rho = 0.8182$. The sample size is $n = 500$. Common random numbers are used across all estimator types. The mse's have standard errors smaller than .004 and the optimal batch size, m^*, are correct to within about one unit of the least-significant digit.



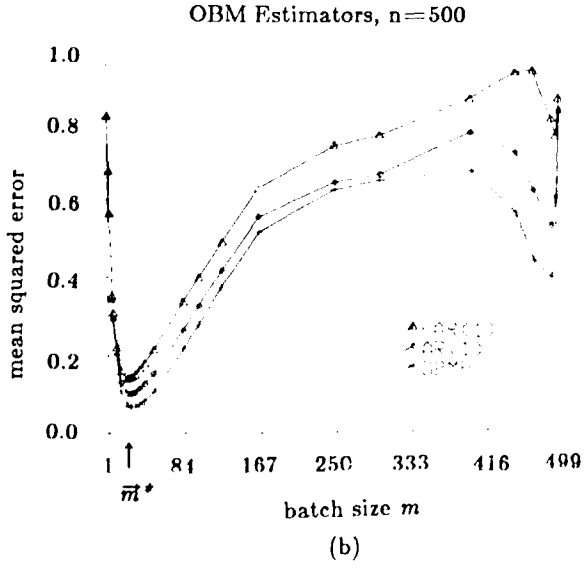
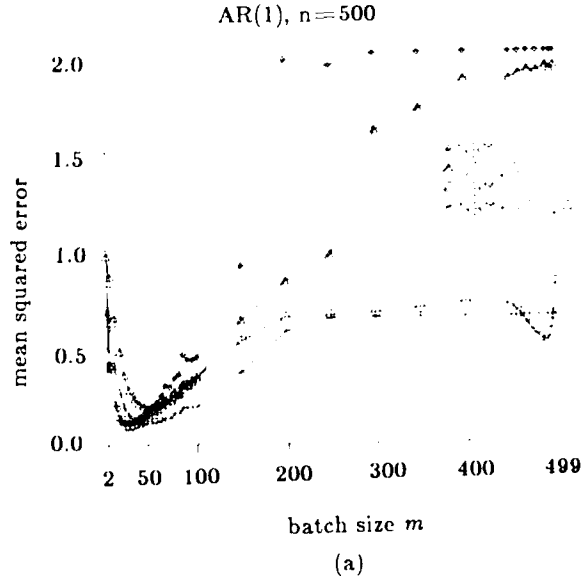


Figure 1. Mse for OBM Estimators applied to EAR(1), AR(1), and S2MC Processes: (a) sample size n=50, (b) sample size n=500.

Figure 2 shows mse as a function of batch size m for NBM, OBM, STS.A, and NBM+STS.A estimators for $n = 500$; Figure 2(a) shows the full range of batch sizes (the feasible ranges of batch sizes for four estimators are listed in Appendix B) and Figure 2(b) zooms in to show batch sizes in the region of minimal mse.



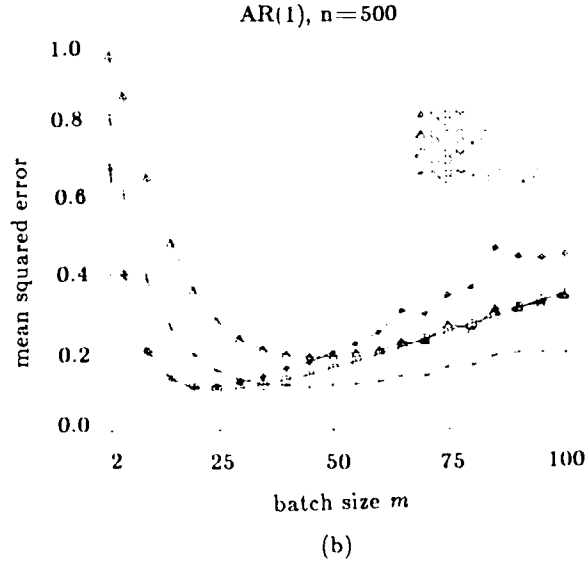


Figure 2. Mse's of NBM, OBM, STS.A, and NBM+STS.A Estimators:
(a)$m = 2, 3, \cdots, 499$, (b)$m = 2, 3, \cdots, 100$.

In this example, for every batch size OBM dominates both NBM and STS.A; also NBM+STS.A dominates STS.A. These results are consistent with the asymptotic values in the last two rows of Table 1.

In this example, the order of the estimators in increasing robustness (second derivative at m^*) are NBM+STS.A, STS.A, OBM, and NBM. This order is consistent with the last row of Table 1. Visually, the values of the finite-sample second derivatives appear consistent with the values of the asymptotic second derivatives.

In this example, the order of the estimators in increasing optimal batch size m^* is NBM, OBM, NBM+STS.A, and STS.A. This ordering is consistent with the asymptotic order in the third-to-last row of Table 1. Moreover, $\bar{m}^*$ has essentially the same value as m^* for each of the four estimators, as shown in the first two rows of Table 3.

Table 3: Optimal Batch Sizes and Optimal Mse's for Finite Samples Compared to the Asymptotic Formulas. Sample Size $n = 500$.

Property	Estimator Type			
	NBM	OBM	STS.A	NBM+STS.A
$\bar{m}^*$	24	27	49	47
m^*	20	25	50	45
$\bar{mse}(\bar{m}^*)$	0.141	0.108	0.293	0.141
$mse(\bar{m}^*)$	0.11	0.10	0.19	0.11
$mse(m^*)$	0.11	0.10	0.19	0.11

In this example, the order of the estimators in increasing values of $mse(m^*)$ is OBM, NBM, NBM+STS.A, and STS.A. This ordering is consistent with the asymptotic values $\bar{mse}(\bar{m}^*)$ shown in the next-to-last row of Table 1. However, except for OBM, the optimal mse, $mse(m^*)$, is significantly lower (35% lower for STS.A) than the asymptotic optimal mse, $\bar{mse}(\bar{m}^*)$, as shown in the last row and the third-to-last row of Table 3.

In this example, the optimal batch size m^* and the asymptotic optimal batch size $\bar{m}^*$ have essentially the same value, so the associated mse's also have the same values, as shown in the last two rows of Table 3. To what extent would we have cared if the batch sizes had not matched? As discussed in the last paragraph of Section 3.2, all that is important is whether $\bar{m}^*$ can indicate a batch size having near-optimal mse for finite sample sizes. Therefore, the important comparison is between the last two rows — $mse(m^*)$ compared to $mse(\bar{m}^*)$, rather than between the last row and the third-to-last row — $mse(m^*)$ compared to $\bar{mse}(\bar{m}^*)$.

4. Summary

In this paper, we study the accuracy of the asymptotic optimal batch size as an approximation for finite-sample cases. We consider four types of estimators of $\text{var}(\bar{X})$ that are parameterized by batch size. We consider three data processes, all have geometrically decreasing correlation structure, but different marginal distributions.

In the examples, the optimal batch size m^* and the asymptotic batch size $\bar{m}^*$ yield essentially the same mse even for sample sizes as small as $n = 50$ and sum of correlations $\gamma_0 = 10$. That is, the asymptotic optimal batch size formula (3.1) worked well in our examples.

Appendix A:

We show here that for processes with geometrically decreasing correlation structure, the relationships discussed in Section 3.1 between ρ, γ_0, and R_0 hold. We also show that the S2MC process obtains the correct mean, $\text{var}(\bar{X})$, and γ_0.

Since

$$\gamma_0 \equiv 1 + 2\sum_{h=1}^{\infty} \rho_h = 1 + 2\sum_{h=1}^{\infty} \rho^h = 1 + 2\left(\frac{\rho}{1-\rho}\right),$$

we have $\rho = \dfrac{\gamma_0 - 1}{\gamma_0 + 1}$.

The steady-state variance of $\bar{X}$ is

$$\begin{aligned}
&\text{var}(\bar{X}) \\
&= \frac{R_0}{n}\left[1 + 2\sum_{h=1}^{n-1}(1 - h/n)\rho_h\right] \\
&= \frac{R_0}{n}\left[1 + 2\sum_{h=1}^{n-1}(1 - h/n)\rho^h\right] \\
&= \frac{R_0}{n}\left[\left(\frac{1+\rho}{1-\rho}\right) - \frac{2\rho(1-\rho^n)}{n(1-\rho)^2}\right],
\end{aligned}$$

so $R_0 = n\,\text{var}(\bar{X})\left[\left(\dfrac{1+\rho}{1-\rho}\right) - \dfrac{2\rho(1-\rho^n)}{n(1-\rho)^2}\right]^{-1}$.

Now consider the S2MC process. Let $\{X_t\}_{t=1}^{n}$ be a dependent symmetric Bernoulli process with parameters μ, $\text{var}(\bar{X})$, and γ_0. Let the state space be $\{c, d\}$ and the transition matrix

$$\mathbf{P} = \begin{bmatrix} (1+\rho)/2 & (1-\rho)/2 \\ (1-\rho)/2 & (1+\rho)/2 \end{bmatrix}.$$

We first show that $\rho_h = \rho^h$ and then show that $c = \mu - R_0^{-1/2}$ and $d = \mu + R_0^{-1/2}$.

Since $\mathbf{P}$ is doubly Markov, at steady state c and d are equally likely (i.e., $\Pr(X_t = c) = \Pr(X_t = d) = 1/2$). Let $Z_t = (d-c)^{-1}(X_t - c)$. Then Z_t has state space $\{0,1\}$ and transition matrix $\mathbf{P}$. At steady state

$$
\begin{aligned}
&\operatorname{cov}(Z_i, Z_{i+h}) \\
&\quad = \mathrm{E}(Z_i\, Z_{i+h}) - \mathrm{E}(Z_i)\mathrm{E}(Z_{i+h}) \\
&\quad = \Big(\mathrm{E}(Z_i\, Z_{i+h} \mid Z_i = 0)\, P(Z_i = 0) \\
&\quad\quad + \mathrm{E}(Z_i\, Z_{i+h} \mid Z_i = 1)\, P(Z_i = 1)\Big) - (\tfrac{1}{2})^2 \\
&\quad = P_{11}^{(h)}(\tfrac{1}{2}) - \frac{1}{4} \\
&\quad = \left(\frac{1+\rho^h}{2}\right)\frac{1}{2} - \frac{1}{4} \\
&\quad = \frac{\rho^h}{4},
\end{aligned}
$$

where

$$
\begin{aligned}
P_{11}^{(h)} &\equiv \Pr(Z_{i+h} = 1 \mid Z_i = 1) \\
&= \Pr(X_{i+h} = d \mid X_i = d) = \frac{1+\rho^h}{2}
\end{aligned}
$$

by induction on h. Therefore,

$$
\operatorname{corr}(Z_i, Z_{i+h}) = \frac{\operatorname{cov}(Z_i, Z_{i+h})}{\operatorname{var}(Z_i)} = \frac{(\rho^h/4)}{(1/4)} = \rho^h .
$$

Since correlation is scale and location invariant,

$$
\rho_h \equiv \operatorname{corr}(X_i, X_{i+h}) = \operatorname{corr}(Z_i, Z_{i+h}) = \rho^h .
$$

Since $X_i = c + (d-c)Z_i$ and the two states are equally probable, the variance is $R_0 = \frac{(d-c)^2}{4}$ and the mean is $\mu = \frac{c+d}{2}$. (The variance is $(d-c)^2$ times the variance of an equally probable Bernoulli trial.) Solving these two equations for c and d yields the results.

Appendix B: Feasible Regions of Batch Sizes

The feasible ranges of batch size m for NBM, OBM, STS.A, and NBM+STS.A are listed below:

NBM: $m = 1, 2, \cdots, \lfloor n/2 \rfloor$,
OBM: $m = 1, 2, \cdots, n-1$,
STS.A: $m = 2, 3, \cdots, n$,
NBM+STS.A: $m = 2, 3, \cdots, \lfloor n/2 \rfloor$, where $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a.

References

[1] U.N. Bhat and Ram Lal, "A sequential inspection plan for Markov dependent production processes," Unpublished Report, Department of Statistical Science, Southern Methodist University, Dallas, Texas, 1988.

[2] M.A. Crane and D.L. Iglehart, "Simulating stable stochastic systems, III: regenerative process and discrete-event simulations," *Operations Research* **23** (1975), 33-45.

[3] G.S. Fishman, Grouping observations in digital simulations, *Management Science* **24** (1978), 510-521.

[4] G.S. Fishman, *Principles of Discrete Event Simulation,* Wiley-Interscience, New York, 1978.

[5] W.A. Fuller, *Introduction to Statistical Time Series.* John Wiley & Sons, 1976.

[6] D. Goldsman and M.S. Meketon, "A comparison of several variance estimators", Technical Report J-85-12, Operations Research Department, AT&T Bell Laboratories, Holmdel, NJ 07733, 1986.

[7] D. Goldsman, "On using standardized time series to analyze stochastic processes", Ph.D. Dissertation, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, 1984.

[8] P. Heidelberger and P.D. Welch, A spectral method for confidence interval generation and run length control in simulation, *Communications of the ACM* **24** (1981), 233-245.

[9] P.A.W. Lewis, Simple models for positive-valued and discrete-valued time series with ARMA correlation structure, *Multivariate Analysis-V,* P.R. Krishnaiah (ed.), North Holland, Amsterdam, 151-166, 1980.

[10] M.S. Meketon and B. Schmeiser, "Overlapping batch means: something for nothing?", *Proceedings of the 1984 Winter Simulation Conference,* S. Sheppard, U. Pooch, and D. Pegden (eds.), 227-230, 1984.

[11] B.W. Schmeiser, Batch-size effects in the analysis of simulation output, *Operations Research* **30** (1982), 556-568.

[12] T.J. Schriber and R.W. Andrews, ARMA-based confidence interval procedures for simulation output analysis, *American Journal of Mathematical and Management Science* **4** (1984), 345-373.

[13] L.W. Schruben, Confidence interval estimation using standardized time series, *Operations Research* **31** (1983), 1090-1108.

[14] B.W. Schmeiser and Tina Song, "Variance Estimators of the Sample Mean: Optimal Mean-Squared Error Batch Sizes," Technical Report 88-15, School of Industrial Engineering, Purdue University, West Lafayette, Indiana, 1988.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. DMS-8717799. The Government has certain rights in this material.

SIMEST AND SIMDAT: DIFFERENCES AND CONVERGENCES

E. Neely Atkinson, Barry W. Brown and James R. Thompson
M.D. Anderson Cancer Center and Rice University

Introduction. Stochastic process modeling has, until fairly recently, exhibited serious limitations in biomathematics, econometrics and other areas of potential application. Consequently, investigators have frequently been driven to linear regression and other ad hoc techniques, with generally poor results. The reason for the difficulty in applied stochastic process modeling is that the axioms, since the time of Poisson, have been forward in time direction, whereas data analytical techniques, such as those based on maximum likelihood are backwards in time direction. Let us consider, for example the following forward axiomitization of cancer progression considered by Bartoszynski, Brown and Thompson (1982).

(1) For each patient, each tumor originates from a single cell and grows exponentially at rate α.

(2) The probability that a tumor of size $Y_j(t)$, not previously detected and removed prior to time t, is detectable in $[t, t + \Delta t]$ is $b\, Y_j(t)\, \Delta t + o(\Delta t)$.

(3) Until the removal of the primary, the probability of metastasis in $[t, t + \Delta t]$ is $a\, Y_0(t)$, where $Y_0(t)$ is the mass of the primary tumor.

(4) The probability of systemic occurrence of a tumor in $[t, t + \Delta t]$ is $\lambda \Delta t + o(\Delta t)$ independent of the prior history of the patient.

Written as they are, in standard Poissonian forward form, the postulates are extremely simple. However, if we attempt to use one of the backwards "closed form" approaches, e.g., maximum likelihood estimation, we are quickly bogged down in a morass of confusion and complexity. For example, in order to use the maximum likelihood approach, we are confronted with the necessity of computing a number of messy terms. We show one of these in (5).

$$(5)\quad P(T_1 = S, T_2 > S) = \int_0^\infty \int_u^\infty e^{W(S-S')} p(t;1)\, p(S';e^{\alpha u})\, x$$

$$[\lambda + ae^{\alpha(t-u)}] \exp[-\lambda(t-u) - \frac{a}{\alpha}(e^{\alpha(t-u)} - 1)]\, x$$

$$H(v(S-S');S';e^{\alpha S'})\, H(v(S-S')\, e^{\alpha S'};u;e^{\alpha(t-u)})\, du\, dt +$$

$$\int_0^\infty \int_u^\infty e^{W(S-S')} p(t;1) \exp[-\lambda t - \frac{a}{\alpha}(e^{\alpha t} - 1)] \lambda e^{-\lambda u}\, x$$

$$p(S'-u;1)\, H(v(S-S');S'-u;1)\, du\, dt.$$

where

$$(6)\quad H(s;t;z) = \exp\{\frac{az}{\alpha} e^{\alpha t}(e^{s} - 1) \log[1 + e^{-s}(e^{-\alpha t} - 1)]$$

$$+ \frac{\lambda s}{\alpha} - \frac{\lambda}{\alpha} \log[1 + e^{\alpha t}(e^{s} - 1)]\},$$

$$(7)\quad p(t;z) = b\, z\, e^{\alpha t} \exp\left[-\frac{bz}{\alpha}(e^{\alpha t} - 1)\right],$$

$$(8)\quad w(y) = \lambda \left[\int_0^y e^{-v(u)} du - y\right],$$

and $v(u)$ is determined from

$$(9) \qquad u = \int_0^v \frac{ds}{a + b + \alpha s - ae^{-s}}$$

The order of computational complexity here is roughly that of four dimensional quadrature. This is near the practical limit of contemporary mainframe computers. The time required for the estimation of the four parameters in the above model was roughly two hours using the robust optimization routine STEPIT on the CYBER 173. A moment's reflection reveals the problem. If a tumor is detected at a particular time, we must examine all possible paths which might have given rise to its origin. For example, it might be a metastasis from the primary, or a metastasis from a metastasis of earlier origin, or a systemically generated tumor, or a metastasis from a systemically generated tumor, etc. Each of these paths is easy to write in the forward direction, but when computing the likelihood, we reason backwards.

In 1983 and 1987, we have presented the algorithm SIMEST for dealing with the backwards/forwards dilemma. In this algorithm, we have returned to the older goodness of fit philosophy of Karl Pearson. Namely, we consider that a set of parameters is close to truth when simulations based on them produce results which appear to be sufficiently similar to those of the data. The procedure is based on binning "failure times" from the data, and noting whether the simulated failure times fall in the bins in a manner similar to those from the data. For example, if we use the formal goodness of fit criterion, we have

$$(10) \qquad S_3(\Theta) = \sum_{j=1}^{l} \frac{(\hat{p}_{lj} - \hat{p}_j)^2}{\hat{p}_j} .$$

where $\hat{p}_{sj}$ is the proportion of simulations falling in the $j'th$ bin and $\hat{p}_j$ is the proportion of actual data points falling in the $j'th$ bin. A major problem with the implementation of an algorithm based on (10) is the fact that fact that, as we proceed from point to point in the parameter space, the criterion will exhibit simulation induced noise. We have addressed this problem in Atkinson, Brown and Thompson (1987) by utilizing a fixed seed approach. Thus, if we are using, say 1,000 simulations at each parameter value, we use the same sequence of seeds for each parameter values. Such an approach enables us, rather than developing stochastic optimization procedures, simply to use existing deterministic software (e.g., that of Nelder and Mead).

In many applications, we will not simply have a one dimensional response variable (e.g., failure time) but a number of response variables. Simple Cartesian binning in such a situation exhibits numerous difficulties, e.g., the empty space phenomenon, namely the fact that most of the bins will be empty of data points. The major purpose of this paper is to address alternatives to Cartesian binning in the multivariable response situation.

Discussion. Let us suppose we have a Poissonian model $M(\Theta)$ for which the vector parameter Θ is unknown, but from which we have a set of n observations of a k-variate response variable $\boldsymbol{X}$. We shall assume a value of Θ and, using this parameter, generate a set of N simulated "data points" $\boldsymbol{Y}$. If the assumed parameter is, in fact, that which generated the actual data, then we should find that the $\boldsymbol{X}$ cloud is indistinguishable from the $\boldsymbol{Y}$ cloud. To determine whether this is so, we might determine the distance of each of the $\boldsymbol{X}$ points from each of the $\boldsymbol{Y}$ points and from each other. Such a matrix of distances should provide all the information we need, but would require $(n + N - 1)n$ elements for each simulation, of which some many thousands will be required in order to find a

satisfactory value of Θ. Accordingly, absent the availability of highly parallel computer architecture with hundreds of CPU's, we need to seek more economical criteria.

We start out with a real world system observable through k- dimensional observations X. We believe that the generating system can be approximately described by a model characterized by the parameter Θ. If we have a data set from the system of size n, we can, for a value of Θ, simulate a quasidata set of size N. Then, we compute the sample mean vector and covariance matrix of the X data set. Then, we transform the data set to a transformed set $\boldsymbol{U} = \boldsymbol{AX} + \boldsymbol{b}$ with mean zero and identity matrix $\boldsymbol{I}$. We then apply the transformation $\boldsymbol{T}$ to the simulated data set. We compute the sample mean $\overline{\boldsymbol{X}}$ and covariance matrix Σ of the simulation data set. If the simulated set is essentially the same as the actual data set, then it should have for its transformed values mean zero and identity covariance matrix $\boldsymbol{I}$. If the underlying data distribution is not too bizarre, we can measure the fidelity of the simulation data to the actual data by computing the ratio of the Gaussian likelihoods of the transformed simulated data sets using the mean and covariance estimates from the actual data and the simulated data respectively. Defining

$$(11) \qquad Q(u_{1i}, u_{2i}, \ldots, u_{ki}) = \sum_{j,l=1}^{k} \sigma^{jl}(u_{ji} - \overline{X}_j)(u_{li} - \overline{X}_l),$$

where σ^{jl} *is the j,l'th* element of the inverse of Σ, we have

$$(12) \qquad L(\Theta) = \frac{\prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{k}{2}}} \exp[-\frac{1}{2}(u_{1i}^2 + u_{2i}^2 + \ldots + u_{ki}^2)]}{\prod_{i=1}^{N} \frac{\sqrt{|\Sigma^{-1}|}}{(2\pi)^{\frac{k}{2}}} \exp[-\frac{1}{2} Q(u_{1i}, u_{2i}, \ldots, u_{ki})]}.$$

We note that (12) involves the computation of only $n + N$ distances. For N large and the underlying distributions Gaussian, the procedure approaches that based on the likelihood ratio test and enjoys its optimality properties. When the underlying distributions are not Gaussian, the procedure is no longer optimal, but will frequently yield the correct value of Θ as $n \to \infty$. We note that it is by no means correct to assume that the test procedure we employ must be based on a consistent nonparametric density estimator. In any event, even when more complex algorithms are required, it will generally be useful to use (12) to move the starting value of Θ closer to truth.

If the distributions of the response variables are very different from Gaussian, it may be appropriate to develop a nonparametric procedure which can become more and more complex until the number of distances required per simulation can approach that of $(N + n - 1)n$. For the first step, we again carry out the transformation $\boldsymbol{T}$ mentioned above which transforms the data set to one with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{I}$.. We record the distances of each of the data points from $\boldsymbol{0}$, say $\{d_j\}_{j=1,n}$, and those of the simulated data points from $\boldsymbol{0}$, say $\{d_{Si}\}_{i=1,N}$. If we have arrived at the true value of Θ, then when the two distance lists are put into one of length $n + N$, we should expect to find an equal distribution of simulated and actual data values throughout the list. Letting W denote the sum of the ranks in the total list of the n data points, we know that if the true value of Θ has been assumed for the simulated data set, we have a standard Wilcoxon-Mann-Whitney situation. Thus, we let

$$(13) \quad U = nN + \frac{n(n+1)}{2} - W;$$

$$\sigma_U^2 = \frac{nN(n+N+1)}{12}; \quad \mu_U = \frac{nN}{2};$$

$$Z = \frac{U - \mu_U}{\sigma_U}.$$

If the value of Θ used in the simulation is the same as that in the model, we know that Z is normally distributed with mean 0 and variance 1. This gives us a natural stopping rule when changing Θ in the optimization routine.

Interestingly, although when a correct value of Θ is assumed, Z must be N(0,1), the fact that Z is N(0,1) does not necessarily guarantee that we have arrived at the correct value of Θ. Note that we made our decision based on relative distances of the data and simulated values from the mean of the data. It is easy to extend this relationship to other data points. For example, we might rank the transformed data distances from **0** and pick, as a second anchor point, the point with median distance from **0**. We then compute the distances of the data points and the simulated data points from this second anchor point. Again, if the correct value of Θ has been picked, our new Z must be N(0,1). Proceeding in this fashion, subsequent anchor points would be those ranked in distance from the data mean, $n/4$, $3n/4$, $n/8$, $5n/8$, $3n/8$, $7n/8$, $n/16$, ...

One difficulty with the above approach is that the test statistics for each anchor point are not stochastically independent. Thus, the significance levels cannot quite been obtained by multiplication of tail area probabilities from the standard normal distribution. In fact, it would be irrelevant to do so anyway, since it is possible to have rank matching about several anchor points without actually having picked Θ correctly. Generally speaking, however, it will be very unusual for rank tests about a few anchor points all to pass satisfactorily the N(0,1) test unless Θ has been correctly selected.

Finally, let us consider the possibility of employing the SIMDAT algorithm in conjunction with SIMEST in order to effect parameter estimation.

Let us suppose we have a random sample $\{X_j\}$ $j=1$ to n of k dimensional vectors. SIMDAT generates pseudo random vectors from the underlying, but unknown distribution that gave rise to the random sample. First of all, we carry out a rough rescaling, so that the variability in each of the k dimensions is approximately equal. We pick an integer m between 1 and n (the method of selecting m will be discussed shortly). For each of the n data points, we determine the $m-1$ nearest neighbors using the ordinary Euclidean metric.

To start SIMDAT, we randomly select one of the n data points. We then have m vectors, the data point selected and its $m-1$ nearest neighbors. The vectors $\{X_j\}$ $j = 1$ to m are now coded about their sample mean

$$(14) \quad \bar{X} = \frac{1}{m} \sum_{i=1}^{m} X_i ,$$

to yield

$$(15) \quad \{X_j'\} = \{X_j - \bar{X}\}_{j=1}^{m}.$$

Next, we generate a random sample of size m from the one dimensional uniform distribution

$$(16) \quad U\left(\frac{1}{m} - \sqrt{\frac{3(m-1)}{m^2}}, \frac{1}{m} + \sqrt{\frac{3(m-1)}{m^2}}\right).$$

This particular uniform distribution is selected to provide the desired moment properties below. Now the linear combination

$$(17) \quad X' = \sum_{l=1}^{m} u_l X_l'$$

is formed, where $(u)l = 1$ *to* m is a random sample from the uniform distribution in (16). Finally, the translation

$$(18) \quad X = X' + \bar{X},$$

restores the relative magnitude, and X is a simulated vector which we propose to be representative of the multivariate distribution that generated the original data set. To obtain the next simulated vector, we randomly select another point from the original data base and repeat the above sequence (sampling with replacement). Although it is very easy
and quick to use, SIMDAT essentially gives the same results that one would obtain by laboriously obtaining the nonparametric probability density estimator and sampling from it.

The selection of m is not particularly critical. Naturally, if we let $m = 1$, we are simply sampling from the data set itself (this is Efron's "bootstrap"), and will experience the difficulties present when one attempts to use a discrete entity to approximate a continuous one. When we use too large a fraction of the total data set, we will tend to obscure fine detail. But the selection of m is not the crucial matter that it is in the area of nonparametric density estimation. Experience indicates that the use of m values in the 5% range appears to work reasonably well.

In the present application, we note the symmetry between SIMDAT and SIMEST. SIMDAT makes no model assumptions beyond continuity of the density function. SIMEST "creates" its own "data" and is completely driven by the model parameter Θ. Using a Pearsonian philosophy, we should select a Θ value which causes the data clouds generated by SIMDAT and SIMEST to be essentially indistinguishable. The means for carrying out this task in a computer efficient fashion is a matter of investigation for us at present. One technique which appears attractive is to generate many SIMDAT and SIMEST data sets of size much smaller than the size of the actual data base and use as a measure of agreement rankings of distances from the common transformed origin.

References

Atkinson, E. Neely, Bartoszynski, Robert, Brown, Barry W. and Thompson, James R., (1983). Simulation techniques for parameter estimation in tumor related stochastic processes, in *Proceedings of the 1983 Computer Simulation Conference,* North Holland: New York, 754-757.

Atkinson, E. Neely, Bartoszynski, Robert, Brown, Barry W. and Thompson, James R., (1983). Maximum likelihood techniques, in *Proceedings of the 44th Meeting of the International Statistical Institute, Contributed Papers,* **2,** 494-497.

Bartoszynski, Robert, Brown, Barry W. and Thompson, James R., (1982). Metastatic and systemic factors in neoplastic progression, in *Probability Models and Cancer* (LeCam, Lucien and Neyman, Jerzy, eds.), New York: North Holland, 253-264. Chandler, J.P. (1969). STEPIT, *Behavioral Science,* **14,** 81.

Efron, Bradley (1979). Bootstrap methods — another look at the jacknife, *Annals of Statistics,* **7,** 1-26.

Nelder, J.A. and Mead, R. (1965). A simplex method for function minimization. *Computational Journal,* **7,** pp. 308-313.

Taylor, Malcolm and Thompson, James R. (1986). A data based algorithm for the

generation of random vectors, *Computational Statistics and Data Analysis,* **4,** 93-101.

Thompson, James R., Atkinson, E. Neely and Brown, Barry W. (1987). SIMEST: an algorithm for simulation-based estimation of parameters characterizing a stochastic process, in *Cancer Modeling,* Thompson, James R. and Brown, Barry W., eds., Amsterdam: Marcel Dekker, 387-415.

ACCELERATION METHODS FOR MONTE CARLO INTEGRATION IN BAYESIAN INFERENCE

John Geweke, Duke University

Abstract

Methods for the acceleration of Monte Carlo integration with n replications in a sample of size T are investigated. A general procedure for combining antithetic variation and grid methods with Monte Carlo methods is proposed, and it is shown that the numerical accuracy of these hybrid methods can be evaluated routinely. The derivation indicates the characteristics of applications in which acceleration is likely to be most beneficial. This is confirmed in a worked example, in which these acceleration methods reduce the computation time required to achieve a given degree of numerical accuracy by several orders of magnitude.

Background

In a statistical model the distribution of a vector of random variables $y_T = (y_1, \ldots, y_T)'$ is assumed to be known up to a vector of parameters $\theta = (\theta_1, \ldots, \theta_k)'$. The model may be expressed by the probability density function whose kernel is the likelihood function $L(y_T|\theta)$, with the functional form of L known and θ unknown. In Bayesian inference the unknown vector of parameters θ is regarded as random, and its distribution conditional on the observed vector y_T is derived. If $\pi(\theta)$ is the prior probability density of the parameters then the conditional distribution of θ is $p(\theta|y_T) \propto L(y_T|\underline{\theta})\pi(\theta)$; $p(\theta|y_T)$ is known as the posterior distribution of θ. Virtually all Bayesian inference problems are cast in the form of determining the expected value of a function of interest $g(\theta)$, under the posterior: $E[g(\theta)|y_T] = \int_\Theta g(\underline{\theta})p(\theta|y_T)d\theta$, Θ being the parameter space.

Among the attractions of Bayesian inference are its provision of a logically consistent approach in complex situations and its incorporation in decision theory (Berger and Wolpert, 1984). There have been very substantial problems in the implementation of methods for Bayesian inference, however: these problems have been approached on a case-by-case basis, with limited development of generic methods; analytical results have been obtained only in a limited set of cases; and computations have been slow, relative to classical methods. The development of analytical generic methods is precluded by the intractability of the integrals in $E[g(\theta)|y_T]$ that emerge in all but the very simplest problems.

The generic problem requires determination of

$$E[g(\theta)|y_T] = \int_\Theta g(\theta)\pi(\theta)L(y_T|\theta)d\theta \cdot \left[\int_\Theta \pi(\theta)L(y_T|\theta)d\theta\right]^{-1}.$$

In applying the model the functional form of $L(y_T|\theta)$ may itself be in doubt. If there are m such models indexed by j, with prior probability π_j, then the posterior probabilities of the models themselves are proportional to $p_j = \pi_j \int_\theta \pi_j(\theta)L_j(y|\theta)d\theta$. In this case $E[g(\theta)|y_T]$, unconditional on model choice, is the average of conditional $E[g(\theta)|y_T]$, weighted by the p_j.

Monte Carlo Integration with Importance Sampling

In Monte Carlo integration with importance sampling a sequence of independent and identically distributed random vectors $\{\theta_i\}_{i=1}^n$ is drawn from an importance sampling density $I(\cdot)$; typically n is on the order of 10^4. Heuristically, the importance sampling density should mimic the posterior density and to this end define the weight function $w(\theta) = p(\theta)/I(\theta)$. The value of $\bar{g}_T = E[g(\theta)|y_T]$ is approximated, numerically, by $g_{n,T} = \Sigma_{i=1}^n g(\theta_i)w(\theta_i)/\Sigma_{i=1}^n w(\theta_i)$. Statisticians have been aware of this approach for over twenty years (Hammersley and Handscomb, 1964). Kloek and van Dijk (1978) conjectured $n^{1/2}(g_{n,T} - \bar{g}_T) \Rightarrow N(0, \sigma_T^2)$ and provided a method of approximating σ_T^2 numerically. This is an important result, for it allows routine evaluation of the numerical accuracy of $g_{n,T}$. It has been further shown (Geweke, 1986) that if $I(\theta) > 0 \; \forall \; \theta \in \Theta$, then $g_{n,T} \to \bar{g}_T$. If in addition $E[w(\theta)] < \infty$, and $E[g^2(\theta)w(\theta)] < \infty$, then $n^{1/2}(g_{n,T} - \bar{g}) \Rightarrow N(0, \sigma_T^2)$, where $\sigma_T^2 = E\{[g(\theta) - \bar{g}_T]^2 w(\theta)\}$; and if $\hat{\sigma}_{n,T}^2 = \Sigma_{i=1}^n [g(\theta_i) - g_{n,T}]^2 w(\theta_i)^2 / [\Sigma_{i=1}^n w(\theta_i)]^2$ then $n\hat{\sigma}_{n,T}^2 \to \sigma_T^2$. (All convergence is in n, the number of Monte Carlo replications.) The conditions for convergence guide the choice of $I(\theta)$, which is an experimental design problem. Over the past two years readily applicable methods for analytical identification of families of importance sampling densities that satisfy the moment conditions have been developed (Geweke, 1986). Algorithmic methods for construction of importance sampling densities within these families have been devised and implemented (Geweke, 1988a), but this work is in an early stage.

Antithetic Acceleration

There are well-known variants on Monte Carlo which accelerate convergence, but until quite recently none has been applied to Bayesian inference. A simple generic method is antithetic acceleration, which uses the technique of antithetic variates introduced by Hammersley and Morton (1956). A pair of identically distributed but negatively correlated vectors, θ_i^A and θ_i^B, are drawn from $I(\cdot)$, and the value of g is approximated numerically by

$$g^*_{n,T} = \Sigma_{i=1}^{n/2}\left[g(\theta_i^A)w(\theta_i^A) + g(\theta_i^B)w(\theta_i^B)\right] \cdot \left\{\Sigma_{i=1}^{n/2}\left[w(\theta_i^A) + w(\theta_i^B)\right]\right\}^{-1};$$

and as before we can compute $\sigma^{*2}_{n,T}$ such that $n\hat{\sigma}^{*2}_{n,T} \to \sigma^{*2}_T$. No matter what the scheme for inducing negative correlation between θ_i^A and θ_i^B, so long as these vectors are drawn from $I(\theta)$ the numerical accuracy of the procedure may be evaluated using this result. In a leading simple class of cases it has been shown (Geweke, 1988b) that $T\sigma^{*2}_T/\sigma^2_T$ coverges almost surely to a finite positive constant as $T \to \infty$, and the expression for the limit indicates that the constant is smaller the more nearly linear is the function g. An immediate implication of this result is that the required number of Monte Carlo iterations to achieve given numerical accuracy relative to the dispersion of the posterior density decreases as sample size increases. This raises the possibility that asymptotic approximations, like those developed by Tierney and Kadane (1986), do not necessarily become preferred on practical grounds as sample size increases.

Grid Acceleration

Antithetic acceleration is but one example of an entire class of extensions of Monte Carlo. In general, an m-tuple $\theta_i^1, \ldots, \theta_i^m$ may be drawn on the i'th Monte Carlo replication, each θ_i^j having probability density $I(\theta)$, θ_i^j independent of θ_k^ℓ if $i \neq k$, but θ_i^k and θ_i^ℓ are in general dependent. The value of $\bar{g}_T = E[g(\theta)]$ is approximated, numerically, by $g_{n,m,T} = \Sigma_{i=1}^n\Sigma_{j=1}^m g(\theta_i^j)w(\theta_i^j)/\Sigma_{i=1}^n\Sigma_{j=1}^m w(\theta_i^j)$. (Antithetic acceleration is the special case in which $m = 2$ with θ_i^1 and θ_i^2 negatively correlated.) It is not hard to show that for fixed m, $n^{1/2}[g_{n,m,T} - \bar{g}_T] \Rightarrow N(0, \sigma^2_{m,T})$, and if

$$\hat{\sigma}^2_{n,m,T} = \Sigma_{i=1}^n\left\{\left[\Sigma_{j=1}^m w(\theta_i^j)g(\theta_i^j)/\Sigma_{j=1}^m w(\theta_i^j) - g_{n,m,T}\right]^2 \left[\Sigma_{j=1}^m w(\theta_i^j)\right]^2\right\} \cdot \left\{\left[\Sigma_{i=1}^n\Sigma_{j=1}^m w(\theta_i^j)\right]^2\right\}^{-1}$$

then $n\hat{\sigma}^2_{n,m,T} \to \sigma^2_{m,T}$. Hence *any acceleration method of this form can be routinely employed.*

The *design* of nonrandom sampling schemes is a question of very great practical importance. Grid or quadrature methods provide one basis for these schemes. To illustrate a simple grid method, let $u^* = (u_1^*, \ldots, u_k^*)'$ be chosen at random from the unit hypercube in $\mathbb{R}^k$, and define an ℓ-grid in $\mathbb{R}^k$ by all points of the form $u = (u_1, \ldots, u_k)'$, $u_j = u_j^* + i/\ell$ $(i = 0, \ldots, \ell-1)$ modulo 1. Map this grid into ℓ^k points in Θ via the inverse c.d.f. of the importance sampling density $I(\theta)$. This provides a feasible method for low-dimensional problems. For higher-dimensional problems, this particular method is impractical because of the rapid increase in ℓ^k. In these problems the grid mesh need not be the same for all parameters; grids may be used for some parameters but not others; or, antithetic variates may be used for some parameters and grids for others.

To provide the intuition for the acceleration inherent in grid methods, consider the motivating problem of computing $\int_0^1 x\,dx = 1/2$. Let $x_i^1 \sim U(0, m^{-1})$ and $x_i^j = x_i^1 + (j-1)/m$, $j=2, \ldots, m$, and denote $\bar{x}_i = m^{-1}\Sigma_{j=1}^m x_i^j$. If the integral is approximated by $\bar{x}_{n,m} = n^{-1}\Sigma_{i=1}^n \bar{x}_i$ then $\mathrm{var}(\bar{x}_{n,m}) = 1/(12m^2n)$. To achieve $\mathrm{var}(\bar{x}_{n,m}) = v^*$, $n = 1/(12m^2v^*)$ Monte Carlo iterations are required. With computation time proportional to mn, computation time required to achieve $\mathrm{var}(\bar{x}_{n,m}) = v^*$ is proportional to $1/(12mv^*)$. This suggests that *required computation time with grid acceleration is approximately inversely proportional to the number of points, that is, approximately proportional to the mesh of the grid.* For the generic case in Monte Carlo integration this conclusion seems a reasonable conjecture, because of the local linearity of inverse c.d.f.'s and functions of interest. For mixtures of grids for some parameters and antithetic variates for others the situation is less clear. Yet another complication is the choice of n: since variance is proportional to n^{-1} and m^{-2}, computational efficiency alone would suggest $n=1$. However, $n > 1$ is required in order to provide $\hat{\sigma}^2_{n,m,T}$ and an assessment of the numerical accuracy of the whole procedure. To explore these practical matters we turn to a worked example.

A Worked Example

We apply these methods to one of the most widely used econometric models, the linear model with first-order autocorrelation:

$$y_t = x_t'\beta + \epsilon_t; \qquad \epsilon_1 \sim N(0, \sigma^2(1-\rho^2)^{-1});$$

$$\epsilon_t = \rho\epsilon_{t-1} + \eta_t, \quad t=2, \ldots, T;$$

$$\eta_t \sim IIDN(0,\sigma^2).$$

The dependent variable is y_t; x_t is a $k \times 1$ vector of independent variables; ϵ_t is an unobserved disturbance; β is a vector of unknown parameters; σ^2 is an unknown positive variance parameter; and the autocorrelation parameter ρ is less than one in absolute value. Letting

$$y_1(\rho) = (1-\rho^2)^{1/2}y_1; \qquad x_{t1}(\rho) = (1-\rho^2)^{1/2}x_{t1};$$

$$y_t(\rho) = y_t - \rho y_{t-1}; \qquad x_{tj}(\rho) = x_{tj} - \rho x_{t-1,j};$$

$$x_t(\rho) = \left[x_{t1}(\rho), \ldots, x_{tk}(\rho)\right]';$$

by a standard transformation (e.g., Theil (1971, pp. 250-253)) the log-likelihood function is

$$-T\log(\sigma) + (1/2)\log(1-\rho^2)$$

$$- (1/2\sigma^2)\Sigma_{t=1}^{T}\left[y_t(\rho) - x_t(\rho)'\beta\right]^2.$$

A standard conjugate prior is $\pi(\beta, \sigma, \rho) \propto \sigma^{-1}$. Conditional on ρ, the posterior in σ is inverse-gamma, and conditional on ρ and σ the posterior in β is multivariate normal. With this in mind, denote the posterior distribution of β and σ conditional on ρ by $\psi(\sigma|\rho)\phi(\beta|\sigma, \rho)$. It is straightforward to sample from $\psi(\cdot)$ and $\phi(\cdot)$. Given an importance sampling density $I^*(\rho)$ for ρ, we may choose the importance sampling density for all parameters to be $I(\theta) = I(\beta, \sigma, \rho) = I^*(\rho)\psi(\sigma|\rho)\phi(\beta|\sigma, \rho)$. Since the range of ρ is limited, if $I^*(\rho) > 0 \; \forall \; \rho \in [-1, 1]$ then $E\left[w(\theta)\right] < \infty$, and for most functions of interest it will be readily verified that $E\left[g^2(\theta)w(\theta)\right] < \infty$. A normal importance sampling density is therefore a reasonable candidate for $I^*(\rho)$.

The posterior mode is found by a global Hildreth-Lu (1960) search in ρ, followed by local maximization with a convergence criterion of 10^{-7} in ρ. For local values of ρ, the posterior was maximized in β and σ, and the log posterior was compared with its value at the global maximum. For each such comparison, a normal density with mean at the global mode may be fit to the two points; the standard deviation of this density is greater, the smaller the difference in the log posterior at the two points. The standard deviation of the importance sampling density is taken to be the largest such standard deviation, over the range for which the difference in log posteriors is less than 20. (Choosing the largest standard deviation is likely to reduce $E\left[w(\theta)\right]$, as discussed in Geweke (1986).)

This procedure determines the importance sampling density $I^*(\rho)$ for ρ. Let $J^*(\rho)$ be the corresponding c.d.f., and $J^{*-1}(\rho)$ the inverse c.d.f. Given a preset number of gridpoints m, draw $u_{i1} \sim U(0, m^{-1})$ and set $u_{ij} = u_{i1} + j/m(j=2, \ldots, m)$; then $\rho_{ij} = J^*(u_{ij})$. We confine the grid to the single parameter ρ. Conditional on ρ, σ and β are independent and conditional on ρ and σ the distribution of β is symmetric. Thus, β appears well suited for antithetic acceleration: if a function of interest is an element of β, then antithetic acceleration provides the exact mean of that element conditional on ρ, and the numerical approximation is exact up the approximation of the unconditional distribution of ρ.

We employ two data sets, each of which is modelled by

$$y_t = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \epsilon_t,$$

where y_t is the log of consumption of a consumer good, x_{t1} is the log of income, and x_{t2} is the log of the price of the consumer good relative to a general price index. In the first example, y_t is log consumption of spirits, and the data are the 69 annual observations used in the example of Durbin and Watson (1951). The least squares coefficients are $b_0 = 4.607$, $b_1 = -.120$, $b_2 = -1.228$; the posterior mode is $\hat{\beta}_0 = 2.453$, $\hat{\beta}_1 = .622$, $\hat{\beta}_2 = -.929$, $\hat{\rho} = .993$. In the second example, y_t is log consumption of textiles, and the data are the 17 annual observations provided by Theil and Nagar (1961). The least squares coefficients are $b_0 = 1.374$, $b_1 = 1.143$, $b_2 = -.829$; the posterior mode is $\hat{\beta}_0 = 1.359$, $\hat{\beta}_1 = 1.149$, $\hat{\beta}_2 = -.827$, $\hat{\rho} = -.125$.

To explore the question of computational efficiency as a function of the number of grid points, six functions of interest were selected: the posterior means of β_1, β_2, and ρ; the predictive mean of a one-period-ahead value of y_t, taking x_{t+1} at its sample mean and ϵ_t at its standard deviation; and $P\left[|\rho| < .1\right]$ and $P\left[\rho > 0\right]$, each computed under the posterior. In the spirits example $P\left[|\rho| < .1\right] = 0$ and $P\left[\rho > 0\right] = 1$, and the latter two functions of interest are not reported. In each case, $n = 2{,}000$. In one set of experiments β was sampled by simple Monte Carlo, and in the other antithetic acceleration for β was employed. Software developed by the author reports the posterior mean, the posterior standard deviation, $\hat{\sigma}^2_{n,m,T}$, and computation time. Given these, computation time required for $\sigma_{n,m,T}$ to fall to one-percent of the posterior standard deviation was computed for the first four functions of interest, and computation time for $\sigma_{n,m,T}$ to fall to .005 was computed for the two probabilities.

Results are reported in Tables 1 and 2. Computation times are given in seconds, using a MicroVax II and double precision arithmetic. (Each valuation of the posterior density for a

different value of ρ requires re-solution of the least squares normal equations.) Not surprisingly, without antithetic acceleration computation times for $E[\beta_1]$ and $E[\beta_2]$ are unaffected. Increased m provides some reduction for the prediction which involves ρ, in the textiles example but not the spirits example. With antithetic acceleration the general pattern of reduction in computing time is the same for all functions of interest. In practical terms, the reduction in time afforded by the combination of grid and antithetic acceleration is quite substantial. The textiles example requires about three minutes with no acceleration; with m = 25, ten seconds suffices to produce n = 20 and meet the chosen numerical accuracy criteria. The spirits example requires about fourteen minutes with no acceleration; with m = 25, twelve seconds suffices to produce n = 20 and meet the accuracy criteria.

As expected, increasing the value of m beyond 25 further reduces computation time. However, the need to have some minimum number of n, the number of Monte Carlo iterations, limits these gains as a practical matter. If one uses a rule of thumb that sets a minimum value of n (here we use n = 20) then nothing is gained by increasing m over the value needed to achieve the required numerical accuracy in the minimum number of Monte Carlo iterations. Put another way, standards for numerical accuracy and the requirement of a minimum number of Monte Carlo replications to assess numerical accuracy reasonably well imply an optimal value for m. Here, that value appears to be about 25, but of course this result is specific to the two examples.

Based on a very simple motivating example, we conjectured that required computation time would be approximately inversely proportional to m. Evidence on this conjecture is provided in Table 3, which reports the product of m and computation time given in Table 2. For the textiles example this product is roughly constant across m. For the spirits example the product declines as m increases. The explanation for this behavior lies in the appeal to a local linear approximation in applying the motivating example to the much more complicated problem worked here. In the spirits example, the normal density $I^*(\rho)$ is centered at $\hat{\rho} = .993$, with a standard deviation of .053. Since the log posterior density declined from its mode to $-\infty$ over an interval of length .007, the grid is poor, indeed, for values of ρ above $\hat{\rho}$, until m is large. This difficulty could be obviated by suitable transformation of ρ; but note that the problem affects only the conjectured relationship of computation time to m -- the validity of the numerical procedure itself and computation of $\hat{\sigma}_{n,m,T}$ are not in question.

Conclusion

The results here suggest the possibility of very substantial gains in computational efficiency from acceleration methods. More investigation is clearly warranted. The foremost problem is that full grids are impractical in more than one dimension: with ℓ grid points in each of k dimensions and a full mesh, computation time is proportional to ℓ^{k-2}/v^*. Hence more sophisticated strategies for grid construction bear investigation. The worked example was tailored to a specific problem, and more generic software is required to learn about appropriate design of grid and antithetic sampling in various models. Among the issues to be investigated are the possibility of algorithmic choice of different grid meshes for different parameters, or axes, to increase efficiency; and the potential additional increase in efficiency afforded by suitable preliminary transformation of parameters. A general proof of the proportionality of computation time to grid mesh would be enlightening, although it is not necessary for the implementation of these methods.

Reduction of computing time from over ten minutes to under ten seconds on desktop machines, as reported here, underscores the fact that innovations in algorithms complement ever faster hardware. These complementarities will increase with innovations in hardware architecture. In particular, grid methods are well adapted to vector or parallel processors, because once the random numbers for each Monte Carlo iteration are chosen, the evaluation of the posterior density and functions of interest at different grid points typically involve precisely the same computations (but with different parameter values). Since vector and parallel architectures are now accessible, this seems an opportune time to pursue the implementation of acceleration methods on these machines in anticipation of their wider availability in the intermediate future.

Table 1

Computation Times, Without Antithetic Acceleration

Textiles example

Function of Interest	m=1	m=2	m=5	m=12	m=25	m=50	m=100
$E(\beta_1)$	202.231	181.796	174.000	173.215	175.402	165.480	169.534
$E(\beta_2)$	202.437	184.914	171.478	160.684	164.977	162.552	156.317
$E(\rho)$	178.036	89.000	39.032	18.375	8.784	3.860*	1.680*
Prediction	193.592	146.717	123.079	116.280	115.735	117.563	111.185
$P(\lvert\rho\rvert < .1)$	181.952	66.712	21.692	8.994	4.236*	2.896*	0.523*
$P(\rho > 0)$	156.567	107.887	18.792	14.066	6.717*	3.021*	2.290*
20 replications					*8.744	*17.458	*34.890

Spirits example

Function of Interest	m=1	m=2	m=5	m=12	m=25	m=50	m=100
$E(\beta_1)$	818.197	803.833	785.122	747.708	727.303	764.308	735.952
$E(\beta_2)$	820.990	806.643	769.044	750.209	803.173	729.340	768.943
$E(\rho)$	443.002	395.034	144.442	25.712	6.163*	1.437*	0.405*
Prediction	911.174	887.308	807.634	755.261	724.942	808.616	787.963
20 replications					*11.164	*22.318	*44.599

Computation times are given in seconds for a MicroVax II using 64-bit arithmetic. Trailing asterisks denote times that imply fewer than 20 Monte Carlo iterations; computation time for 20 iterations is indicated by the figure with leading asterisk at the bottom of the column.

Table 2

Computation Times, With Antithetic Acceleration

Textiles example

Function of Interest	m=1	m=2	m=5	m=12	m=25	m=50	m=100
$E(\beta_1)$	1.606	0.959	0.409*	0.191*	0.088*	0.033*	0.012*
$E(\beta_2)$	12.072	12.829	9.279	6.080	4.879*	3.406*	2.121*
$E(\rho)$	174.867	91.394	40.863	18.355	9.447	4.161*	1.802*
Prediction	60.502	27.948	9.935	4.063*	1.787*	0.694*	0.265*
$P(\lvert\rho\rvert < .1)$	173.620	70.353	22.259	9.649	4.226*	3.051*	0.616
$P(\rho > 0)$	151.789	113.245	20.698	13.882	6.856*	3.279*	2.385*
20 replications			*1.855	*4.415	*9.163	*18.295	*36.564

Spirits example

Function of Interest	m=1	m=2	m=5	m=12	m=25	m=50	m=100
$E(\beta_1)$	55.486	41.986	13.208	2.369*	0.592*	0.140*	0.039*
$E(\beta_2)$	51.896	41.185	11.833	2.097*	0.515*	0.117*	0.034*
$E(\rho)$	475.000	427.113	143.177	25.872	6.550*	1.495*	0.424*
Prediction	5.045	4.138	1.598*	0.342*	0.092*	0.023*	0.006*
20 replications			*2.348	*5.567	*11.570	*23.092	*46.182

Computation times are given in seconds for a MicroVax II using 64-bit arithmetic. Trailing asterisks denote times that imply fewer than 20 Monte Carlo iterations; computation time for 20 iterations is indicated by the figure with leading asterisk at the bottom of the column.

Table 3

Computation Time Scaled by Grid Size, With Antithetic Acceleration

Textiles example

Function of Interest	m=1	m=2	m=5	m=12	m=25	m=50	m=100
$E(\beta_1)$	1.606	1.918	2.045	2.287	2.203	1.640	1.187
$E(\beta_2)$	12.072	25.658	46.394	72.966	121.967	170.306	212.078
$E(\rho)$	174.867	182.788	204.317	220.265	236.187	208.068	180.196
Prediction	60.502	55.895	49.673	48.752	44.683	34.695	26.544
$P(\lvert\rho\rvert < .1)$	173.620	140.706	111.293	115.793	105.658	152.572	61.632
$P(\rho > 0)$	151.789	226.489	103.492	166.585	171.402	163.939	238.467

Spirits example

Function of Interest	m=1	m=2	m=5	m=12	m=25	m=50	m=100
$E(\beta_1)$	55.486	83.973	66.039	28.429	14.799	6.996	3.895
$E(\beta_2)$	51.896	82.371	59.164	25.169	12.864	5.858	3.385
$E(\rho)$	475.000	854.227	715.884	310.467	163.748	74.763	42.437
Prediction	5.045	8.275	7.990	4.101	2.304	1.130	0.569

Figures given are the product of those in Table 2, with the corresponding value of m.

References

Berger, J., and R. Wolpert, 1984, *The Likelihood Principle*, Hayward, CA: The Institute of Mathematical Statistics.

Durbin, J., and G.S. Watson, 1951, "Testing for Serial Correlation in Least Squares Regression, Part II," *Biometrika* 38:159-178.

Geweke, J., 1986, "Bayesian Inference in Econometric Models Using Monte Carlo Integration," manuscript.

Geweke, J., 1988a, "Exact Inference in Models with Autoregressive Conditional Heteroskedasticity," in E. Berndt, H. White, and W. Barnett (eds.), *Dynamic Econometric Modeling*, Cambridge: Cambridge University Press, forthcoming.

Geweke, J., 1988b, "Antithetic Acceleration of Monte Carlo Integration in Bayesian Inference," *Journal of Econometrics* 38:73-90.

Hammersley, J.M., and D.C. Handscomb, 1964, *Monte Carlo Methods*. London: Methuen. (First edition)

Hammersley, J.M., and K.W. Morton, 1956, "A New Monte Carlo Technique: Antithetic Variates," *Proceedings of the Cambridge Philosophical Society* 63:449-475.

Hildreth, C., and J.Y. Lu, 1960, "Demand Relationships with Autocorrelated Disturbances," Michigan State University Agricultural Experiment Station Technical Bulletin 276, East Lansing, Michigan.

Kloek, R., and H.K. van Dijk, 1978, "Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo," *Econometrica* 46:1-19.

Theil, H., 1971, *Principles of Econometrics*, New York: John Wiley and Sons.

Theil, H., and A.L. Nagar, 1961, "Testing the Independence of Regression Disturbances," *Journal of the American Statistical Association* 56:793-806.

Tierney, L., and J.B. Kadane, 1986, "Accurate Approximations for Posterior Moments and Marginal Densities," *Journal of the American Statistical Association* 81:82-86.

MIXTURE EXPERIMENTS AND FRACTIONAL FACTORIALS USED TO TAILOR COMPUTER SIMULATIONS

Turkan K. Gardenier, TKG Consultants, Ltd.

ABSTRACT

Large scale computer simulations are in widespread and growing use in government, business and science. Within the Department of Defense the use of simulation is particularly crucial because the real-world scenario of the battle cannot be replicated. Environmental and health simulations for risk assessment have complex determinants of pollution and target sites. Large number of parameters may initially appear to be needed in simulations, and experiment designs achieved through response surface methodology, can reduce the final set parameters to an efficient minimum.

This paper presents the use of several experiment design procedures, including fractional factorials, mixture experiments with contrained optimization and Hadamard matrices as pre-processors to computer simulations. These methods have been used by the author to (a) minimize the number of computer runs, (b) conduct an input-output analysis of model subroutines and measures of merit, (c) check for computational model validity, (d) design interactive graphical evaluation schemes for the simulation developer and user. The use of experiment designs as preprocessors resulted in cost-savings as well as efficient interpretations for battle management.

INTRODUCTION

As the number of parameters increase in computer simulations, the direct or indirect relationship between input and output becomes difficult to quantify. The necessary costs to run the simulation model increases in parallel.

To a statistician, experiment design as a discipline seems the most natural way to approach a screening effort for relevant variables. A simulation setting is the most natural context for collecting the relevant data, analyzing and interpreting them without having to defend "missing cells." In a previous report Gardenier (1982) illustrated the use of statistical principles in the study of complex relationships among simulation input variables. Since then, considerable emphasis has been placed on formulating surrogate models or metamodels--"models" of simulation models--which reduce the input-output relationships to the framework of a regression equation (Friedman, et. al, 1984; Kleijnen, 1982). Biles (1979) also stressed the importance of using statistical principles in designing simulation runs and interpreting the output of simulation experiments.

The objective of the present paper is to:

(a) demonstrate how the principles of experiment design, multivariate data analysis and optimization techniques can be applied to simulation models;

(b) show relative efficiencies among several experiment design plans.

Two ways of structuring statistical experiment designs as pre-processors, an integrated tool denoted as Pre-Prim by the author, will be demonstrated. The first deals with independent input vector parameters; the second deals with constrained mixture experiments.

PRE-PRIM AS A NEW CONCEPT

Pre-Prim, as an integrated set of statistical tools, offers the capability of mechanizing the decisions of the simulation user. It offers the feasibility to:

(a) analyze a maximum number of input parameters with a minimum number of simulation runs;

(b) incorporates non-linearities and synergies of input variables into the mathematical regression model;

(c) assures stability and minimum variance in the coefficients of the metamodel or surrogate model.

Pre-Prim also offers a protocol for sequencing the simulation runs. Thus, if trials were to be interrupted at certain nodal points during the sequence of total experimentation, there would be a minimal impact on parameter efficiency.

Pre-Prim works interactively with the user in order to formulate:

(a) the minima/maxima of the input variables which, in essence, determine the region of the response surface explored;

(b) the nature of the function relating input to output; i.e., whether the relationship can be represented by a linear function, second order or higher order polynomial;

(c) whether 2, 3, or higher levels should be associated with the input variables, as decided upon in point (b) above;

(d) what mode and pattern of interactions among input variables need to be explored.

The total number of input parameters, consisting of main effects or linear terms, interactions, and non-linear terms, determine the design matrix. The design matrix itself determines the degrees of freedom available to estimate the error variance in the multivariate regression metamodel or surrogate model. For statistical efficiency purposes, it is essential to formulate these criteria prior to starting simulation runs which estimate model sensitivity. The pre-processor design matrix needs to maintain the criteria of balance and orthogonality.

PROTOTYPE DESIGN PLANS

Pre-processing design plans can be categorized by the number of "levels" in input variables and their particular mix. Pre-Prim has grouped them into the following categories:

(a) 2 or 3-level screening designs estimating main effects only;

(b) 2 or 3-level designs estimating interactions as well as main effects;

(c) mixed level designs combining 2 and 3-level input variables;

(d) designs involving a constrained sum of preference decisions, based upon mixture experiments.

The first three types assume that the input variables relate as independent vectors to the output. The last type of designs solve for the optimal preference mix in inputs.

Full factorial designs estimate all main effects and all possible interactions up to order k-1 (k refers to the number of variables). The total number of necessary simulation runs in a full factorial design is represented by l^k, where l corresponds to the number of levels.

Fractional factorial designs reduce the simulation run demands to a fraction of what would be needed for a full factorial trading, in return for estimable parameters, economy in computer run cost. Most main effects, and some of the most important 2 or 3-factor interactions are estimated, the choice determined through user interface in Pre-Prim. An example of this type of design, as presented in Brouse and Gardenier (1987) is shown in Table 1.

If no interactions, but only linear main effects are to be explored, it is possible to use the principles of Hadamard matrices and estimate the effects of up to k-1 input variables with as few as k simulation runs. These designs are denoted as "screening designs" because they allow for no interactions. They use only two levels for each factor, a feasible minimum and maximum. Thus they do not allow for estimation of non-linearities.

Pre-Prim includes several plans which allow for the estimation of non-linearities but which also result in cost-savings similar to those offered by fractional factorials and Hadamard matrices. Table 2 shows an illustration of the variable/level combinations in these plans. For example, in line 3 we see that a total of 9 factors can be screened with as few as 8 simulation runsIn this plan, 7 variables have 2 levels, one variable has 3 levels (thus allowing for non-linearity estimation) and one variable has 4 levels. 4-level variables may involve categorical data such as types of aircraft.

In the example above, if we had used a full factorial design estimating only linear effects for each variable, 512 simulation runs would have been required; Hadamard matrices could have reduced the sample size to 10. Both of the above plans would have stimated only linear effects; factorials would have given the full set of interactions, Hadamard matrices no interactions. The special Pre-Prim plans shown in Table 2 require only 8 simulation runs and enable non-linear effect estimation for 2 of the 9 variables.

In constrained sum designs solving for optimal preferences among input variables, the effect of 3 variables can be determined with 7 simulation runs, the effect of 4 input factors with 15 runs, and the effect of 5 variables can be determined with 21 runs. An example of this application is shown below.

AN APPLICATION TO MAN-IN-THE-LOOP BATTLE MANAGEMENT

Let us consider the case of a simulation model where various reentry vehicle (RV) and platform characteristics are being analyzed as to their effect at various phases of the battle: (a)boost, (b) post-boost, (c) midcourse and (d) terminal. After reviewing where each relevant subroutine impacts the output parameters (Kubeja, 1987), each input is related to nodes in the battle-phased output. Figure 1 shows these results. For example, platform characteristics affect all nodal phases; RVs, Target Type and Time of RV Impact affect boost, midcourse and terminal phases respectively. The Target Types impacted also had 5 options:

(1) missile silos;
(2) C^3 sites;
(3) bomber bases;
(4) other military targets;
(5) urban industrial locations.

For a first stage analysis, three RV and two platform characteristics shown in Figure 1 were chosen. A second stage analysis was then formulated for the five options of Target type, holding all other first stage variables constant. Our decision was reached after considering various options for pre-processor design. These, and the associated simulation run requirements, are shown in Table 3.

The two-stage constrained sum design selected represents less than 1/1000 of the simulation run requirements of option IA, only about 4% of option IB, 6% of option II, and 17% of option III. Savings in computer run time and related data interpretation are substantial.

Results for the design selected can be analyzed in two ways. The first is the regression-oriented approach where the input design matrix is submitted to multivariate regression and analysis of variance, ANOVA. The coefficients obtained by matrix inversion are (a) scanned for statistical significance, (b) regression is implemented again, keeping the significant variables, (c) the coefficients are used as the terms of the metamodel. Hypothetical data have been analyzed by this procedure and are shown in Table 4. The results show the hypothetical output using 5 input variables from the 5 input variables in the Stage I mixture design regressed against percentage total leakage during the battle. All input variables

are statistically significant, with confidence coefficients, $1-\alpha$, ranging from .82 to .99.

These results are not as informative as the query of what mix among input variables optimize the criterion output. For example, one question in battle management is to solve for the number of decoys which are optimal for a specific attack scenario; another is the differential benefits of the use of maneuvering versus electronic countermeasures, ECM. In the present application, the query was the optimal mix of preferences in the utility function scale ranging from 1 to 10. A prototype analysis of hypothetical results is shown in Table 5.

In this example, the sum of the weight preferences was set to 20. The optimal mix which would minimize the leakage of attacking RVs is shown in the first column under "value at Minimum." The optimization module was successful in maintaining leakage during battle at less than .0001. The appropriateness of model fit was checked by a plot of residuals or deviations from the response surface.

Another tool for evaluating the response surfaceobtained from the analysis is triad plots (Scheffe, 1963.) Figure 2 includes two hypothetical results for prototype diagrams. These can assist the simulation users in deciding among various alternatives. The letters in the triad plots refer to values of the output criterion variable, leakage, held at values .00 - .75 in intervals of .15. Individual codes are shown next to subfigure I. The vertices in each triad correspond to one of the five alternative inputs. Three vertices are shown, two variables are held constant. At the corner of each vertex, maximal weight is apportioned to that variable.

Evaluating sub-figures I and II we note that it is more beneficial to choose the strategy of sub-figure II. In this hypothetical dataset, as we increase the relative weighting scheme to Platform Resources, leakage values approach zero; we see more C-coded values in contrast to the D- and E-coded values we noted in sub-figure I. The essential difference between sub-figures I and II is that Target Type became a player in subfigure II, replacing Number of RV's in subfigure I.

Interactive graphics of this type, combined with pre-processors and surrogate modeling-related analytical techniques, can aid man-in-the loop related strategy decisions.

CONCLUDING REMARKS

This paper has demonstrated how statistical experiment design principles, used as a Pre-Prim interface to large-scale simulations, can drastically reduce the simulation run costs. Regression oriented surrogate modeling or metamodeling can mathematically represent the relationship between input and output variables. Graphical techniques and optimization algorithms can solve for the best mix among strategies, thus helping the user in tradeoff decisions.

ACKNOWLEDGEMENTS

I am grateful to IIT Research Institute for initiating my interest in the use of statistical experiment design principles to multivariate regression oriented studies. Many of the application contexts to simulation models were provided by ANSER Corporation. Some of the applications were sponsored, in part by the Strategic Defense Initiative Organization contract MDA 903-85-C-0049.

REFERENCES

Gardenier, T. K., "Some Uses of Statistics in Simulation," in Computer Modeling and Simulation: Principles of Good Practice. J. McLeod (ed), La Jolla, CA: Society for Computer Simulation, 1982, pp. 129-139.

Friedman, L. W., and Friedman, H. H., "Statistical Considerations in Computer Simulation: The State of the Art," J. Stats. Computer Simulation, Vol. 9, 1984, pp. 237-263.

Kleijnen, Jack P. C., "Regression Metamodel Summarization of Model Behavior," Encyclopedia of Systems and Controls, M. G. Singh (ed), Oxford: Pergamon Press, 1982, pp. 1-21.

Biles, W. F., "Experimental Designs in Computer Simulation," Proceedings of the 1979 Winter Simulation Conference, LaJolla, CA: Society for Computer Simulation, pp. 3-9.

Brouse, D., and Gardenier, T. K., "Regression Metamodels for Strategic Defense Simulation Analysis," presented at the 1987 Summer Computer Simulation Conference, Montreal, Canada, August, 1987.

Kubeja, K., "The Blue Defender Model Utility Function," presented at the 1987 Summer Computer Simulation Conference, Montreal, Canada, August, 1987.

Scheffe, H., "The Simplex Centroid Design for Experiments with Mixtures," J. Royal Stat. Soc., Ser. B, 1963, pp. 235-263.

TABLE 1
PROTOTYPE FRACTIONAL FACTORIAL DESIGN
USING SIX INPUT FACTORS AND
EIGHT TWO-FACTOR INTERACTIONS

	(1) PLATFORMS	(2) THREAT OBJ	(3) P(KILL)	(4) P(TRACK)	(5) WEAPON RANGE	(6) ACTIVATION TIME	INTERACTIONS							
TRIAL SEQUENCE	X_1	X_2	X_3	X_4	X_5	X_6	X_1X_2	X_1X_3	X_1X_4	X_1X_5	X_1X_6	X_2X_3	X_2X_4	X_2X_5
1	-	-	-	-	-	-	+	+	+	+	+	+	+	+
2	+	-	-	-	-	+	-	-	-	-	+	+	+	+
3	-	+	-	-	-	+	-	+	+	+	-	-	-	-
4	+	+	-	-	-	-	+	-	-	-	-	-	-	-
5	-	-	+	-	-	+	+	-	+	+	-	-	+	+
6	+	-	+	-	-	-	-	+	-	-	-	-	+	+
7	-	+	+	-	-	-	-	-	+	+	+	+	-	-
8	+	+	+	-	-	+	+	+	-	-	+	+	-	-
9	-	-	-	+	-	+	+	+	-	+	-	+	-	+
10	+	-	-	+	-	-	-	-	+	-	-	+	-	+
11	-	+	-	+	-	-	-	+	-	+	+	-	+	-
12	+	+	-	+	-	+	+	-	+	-	+	-	+	-
13	-	-	+	+	-	-	+	-	-	+	+	-	-	+
14	+	-	+	+	-	+	-	+	+	-	+	-	-	+
15	-	+	+	+	-	+	-	-	-	+	-	+	+	-
16	+	+	+	+	-	-	+	+	+	-	-	+	+	-
17	-	-	-	-	+	+	+	+	+	-	-	+	+	-
18	+	-	-	-	+	-	-	-	-	+	-	+	+	-
19	-	+	-	-	+	-	-	+	+	-	+	-	-	+
20	+	+	-	-	+	+	+	-	-	+	+	-	-	+
21	-	-	+	-	+	-	+	-	+	-	+	-	+	-
22	+	-	+	-	+	+	-	+	-	+	+	-	+	-
23	-	+	+	-	+	+	-	-	+	-	-	+	-	+
24	+	+	+	-	+	-	+	+	-	+	-	+	-	+
25	-	-	-	+	+	-	+	+	-	-	+	+	-	-
26	+	-	-	+	+	+	-	-	+	+	+	+	-	-
27	-	+	-	+	+	+	-	+	-	-	-	-	+	+
28	+	+	-	+	+	-	+	-	+	+	-	-	+	+
29	-	-	+	+	+	+	+	-	-	-	-	-	-	-
30	+	-	+	+	+	-	-	+	+	+	-	-	-	-
31	-	+	+	+	+	-	-	-	-	-	+	+	+	+
32	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Table 2

Nonlinear Preprocessor Run Requirements in Some Pre-Prim Designs

Simulation Runs	Variables (V)/ Levels (L)						Total Variables
	V	L	V	L	V	L	
9	4	3	4	2			8
18	7	3	7	2			14
8	1	4	1	3	7	2	9
16	5	4	5	3	15	2	25
32	9	4	9	3	31	2	49

Figure 1

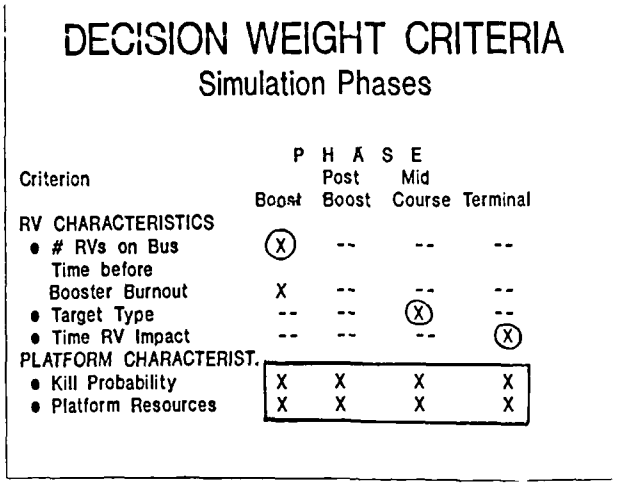


Table 3

Possible Alternative Pre-Processor Designs for Man-in-the Loop

Pre-Processor Alternative	Number of Simulation Runs
I. FULL FACTORIAL: 2 STAGES	
A. 3-level variables	$(3)^{10}$ = 177,147
B. 2-level variables	$(2)^{10}$ = 1,024
II. FULL FACTORIAL AND CONSTRAINED SUM	$(2^5 \times 21)$ = 672
III. HADAMARD SCREENING AND CONSTRAINED SUM	$(1^1 \times 12)$ = 252
IV. TWO-STAGE CONSTRAINED SUM	$(2^2 \times 21)$ = 42

Table 4

Regression Coefficients for LEAKAGEPERCENT

Coefficient	Term	Standard Error	T-Value	Confidence Coef <> 0
0.2998	RVNUMBER	0.1764	1.700	89.5%
0.5780	TARGETTYPE	0.1764	3.277	99.7%
0.7219	IMPACTTIME	0.1764	4.092	99.9%
0.9697	KILLPROB	0.1764	5.497	99.9%
0.2520	PLATFORMRESOUCE	0.1764	1.429	82.4%

Confidence figures are based on 16 degrees of freedom

Analysis of Variance for LEAKAGEPERCENT

Source	df	SS	MS	F-Ratio
Total (corrected)	20	1.3001		
Regression	4	0.4980	0.12451	2.484 (1)
Residual	16	0.8021	0.05013	

(1) Implies 91.4% confidence regression equation is nonzero.

Table 5

MINIMUM LEAKAGEPERCENT

A minimum of 0.000073 was achieved under the following conditions.

Value at Minimum	Factors	Lower Limit	Upper Limit
0.484	RVNUMBER	0.000	1.000
0.609	TARGETTYPE	0.000	1.000
0.466	IMPACTTIME	0.000	1.000
0.0468	KILLPROB	0.000	1.000
0.778	PLATFORMRESOUCE	0.000	1.000

LEAKAGEPERCENT.RES

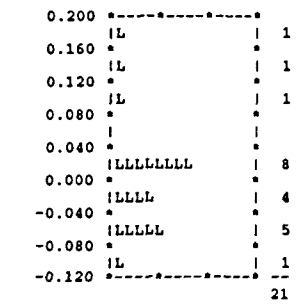
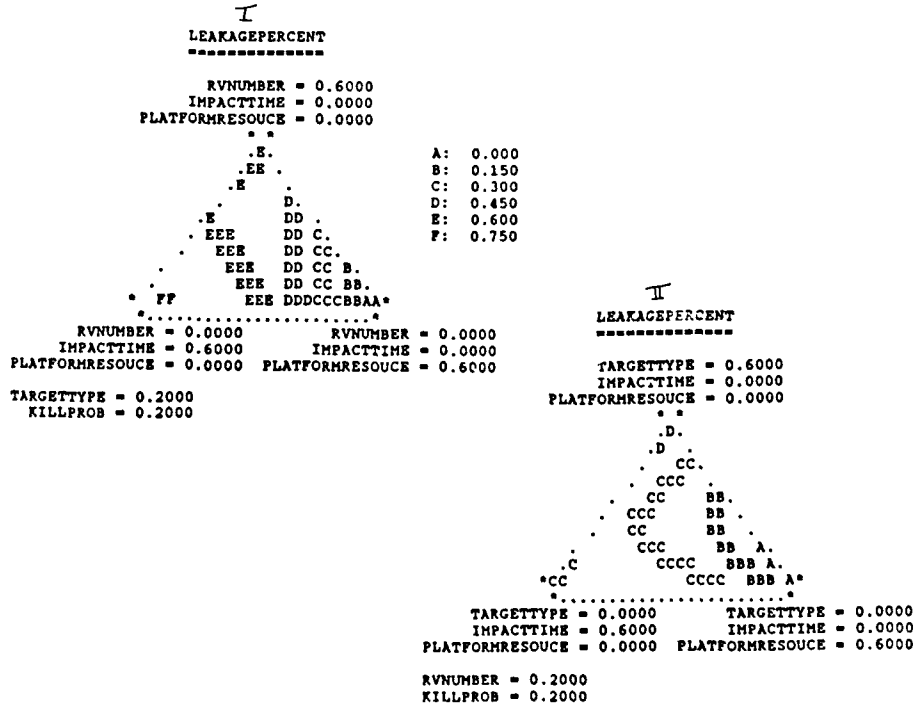


Figure 2
Prototype Diagram of Hypothetical Data Used to Illustrate Use of Triad Contours



SIMULATION AND STOCHASTIC MODELING FOR THE SPATIAL ALLOCATION OF MULTI-CATEGORICAL RESOURCES

Richard S. Segall, University of Lowell, Lowell, MA 01854

I. Background

This paper extends a mathematical model called DRAM (Disaggregated Resource Allocation Model) which was formulated by Venedictov et al. (1977) and later refined by Gibbs (1978) and Hughes (1980) at IIASA (International Institute for Applied Systems Analysis) in Laxenburg, Austria.

Even though the DRAM model was a product of the Health Care Systems Modeling Task Force at IIASA, its applicability to other types of resources is unlimited. Basically, the DRAM model is a simulation model which predicts how a large-scale capacity system with constraints on supply would respond when resource availability changes.

Mayhew (1981a) further extended the DRAM model to account for multi-specialty modeling of patient flows over a geographical region with a model called DRAMOS (Disaggregated Resource Allocation Model Over Space). The DRAMOS model is really a hybrid model between DRAM and a model called RAMOS (Resource Allocation Model Over Space) which was developed and successfully tested by Mayhew (1980a, 1980b) to model single category flows as an aggregate over geographical regions in England and other countries with capacity constraints.

Segall (1982, 1983, 1984, 1987a, c) and Rising et al. (1984a) further extended the RAMOS model and successfully applied it to actual data for the State of Massachusetts as representative of a large scale system which is not capacity constrained, but rather is affected by market forces.

This paper intends to refine the DRAMOS model for application to market systems by assuming a demand constraint on the origin of the consumer instead of a supply constraint for the place of economic consumption. This is analogous to the distinction between the destination and origin constrained forms of the RAMOS model as formulated by Mayhew (1980a). Additionally, probabilistic assumptions are made on certain parameters of the modeling to make a stochastic nature of its applicability possible.

II. Mathematical Modeling

A. *An Origin Constrained DRAMOS Model*

Below is an origin constrained formulation of the DRAMOS model which is an extension of the Mayhew (1981a) destination constrained model.

Define:

i = origin zone ($1 \le i \le m$)

j = destination zone ($1 \le j \le n$)

k = specialty category ($1 \le k \le p$)

T_{ijk} = flow from origin i to destination j in category k

ℓ_{ik} = average length of stay for category k from origin i

W_{ik} = demand from origin i of category k

S_i = total capacity demanded by origin i

Φ_{ijk} = maximum flow from origin i to destination j in category k

L_{ik} = maximum length of stay for category k from origin j

Model Objectives:

To evaluate the values of T_{ijk} and ℓ_{ik} that satisfy the following equations (1) and (2) subject to constraints (3) and (4):

$$W_{ik} = \sum_j T_{ijk} \tag{1}$$

$$\sum_k W_{ik} \ell_{ik} = S_i \tag{2}$$

$$0 \le T_{ijk} \le \Phi_{ijk} \tag{3}$$

$$0 \le \ell_{ik} \le L_{ik} \tag{4}$$

Below are deterministic and stochastic versions of a non-linear preference function originally developed by Hughes (1980) for the consumer's zone i of residence. The optimization problem is formulated with either of these two versions of the objective function subject to the constraints given by equations (3) and (4) above.

The first version is used when the service and demand benefit functions are known precisely. This situation is modeled below by equations (6) and (7), and requires the knowledge of an immense amount of parameters; which is usually not the case. The second version overcomes this difficulty by allowing both the preassigning of parameter values as well as the probabilities of parameter values. The latter version is useful for planning of large scale systems when parameter values are subject to change over the planning horizon rather than being held fixed.

Define:

α_{ik} = relative importance parameter of servicing maximum flow of specialty k from origin i ($\alpha_{ik} > 0$).

γ_{ik} = relative importance parameter of having maximum demand for specialty k from origin i ($\gamma_{ik} > 0$).

$g(T_{ijk})$ = service benefit function

$h(\ell_{ik})$ = demand benefit function

C_i = marginal unit cost of demand in each origin zone i

VERSION 1: DETERMINISTIC NONLINEAR OBJECTIVE FUNCTION

$$U_i(T,\ell) = \sum_{kj} g(T_{ijk}) + \sum_{kj} T_{ijk}\, h(\ell_{ik}) \tag{5}$$

where

$$g(T_{ikj}) = \frac{-\phi_{ijk}\, C_i\, L_{ik}}{\alpha_{ik}} \left(\frac{T_{ijk}}{\phi_{ijk}} \right)^{-\alpha_{ik}} \quad (6)$$

$$h(\ell_{ik}) = \frac{L_{ik}}{\gamma_{ik}} \left[1 - \left(\frac{\ell_{ik}}{L_{ik}} \right)^{-\gamma_{ik}} \right] \quad (7)$$

Analogous supply driven expressions for the above benefit functions of equations (6) and (7) can be found in Gibbs (1978, p.8-9).

VERSION 2: STOCHASTIC NONLINEAR OBJECTIVE FUNCTION

It should be recognized that realistic modeling requires integer values for the variable T_{ijk}, which counts the number of consumers migrating from i to j for commodity or service type k. That is, T_{ijk} is a discrete variable as given by

$$T_{ijk} = 0, 1, 2, \ldots, n \quad (8)$$

We can assign probabilities for the values of T_{ijk} as being equal to each of these integer values, by the following

$$P\left[T_{ijk} = 0, 1, 2, \ldots, n \right] = \sum_{\ell=0}^{n} P\left[T_{ijk} = \ell \right] \quad (9)$$

$$= \sum_{\ell=0}^{n} P_{ijk\ell} \quad (10(a))$$

where $0 \le P_{ijkl} \le 1$ and $\sum\sum\sum\sum_{ijkl} P_{ijkl} = 1.0$. (10(b))

Similarly we can extend a probabilistic interpretation for the nonlinear objective function by taking probabilities of both sides of equation (5):

$$P\left[U_i(T, \ell)\right] = P\left[\sum\sum_{kj} g(T_{ijk})\right] + P\left[\sum\sum_{kj} T_{ijk}\, h(\ell_{ik})\right] \quad (11)$$

$$= \sum\sum_{kj} P\left[g(T_{ijk}) \right] + \sum\sum_{kj} P\left[T_{ijk}\, h(\ell_{ik}) \right] \quad (12)$$

Using equation (6),

$$P\left[g(T_{ijk}) \right] = P(\phi_{ijk}=A) \cdot P(C_i=B) \cdot P(L_{ik}=C) \cdot P(\alpha_{ik}^{-1}=D) \cdot P(\phi_{ijk}^{-\alpha_{ik}}=E) \cdot P(T_{ijk}^{-\alpha_{ik}}=F) \quad (13)$$

where A,B,C,D,E, and F are prespecified values for given i,j, and k.

Taking summations of equation (13) yields:

$$\sum\sum_{jk} P\left[g(T_{ijk}) \right] = \sum\sum_{jk} P_{A_{jk}} \cdot P_{B_i} \cdot P_{C_k} \cdot P_{Dk} \cdot P_{E_{jk}} \cdot P_{F_{jk}} \quad (14)$$

Because usually the values for ϕ_{ijk}, L_{ik} and α_{ik} will be given assumptions of the problem; their associated respective probabilities would be 1.0 and hence further simplification of equation (14) would be possible.

Using equation (7),

$$P\left[T_{ijk} \cdot h(\ell_{ik}) \right] = P(T_{ijk}=G) \cdot P(h(\ell_{ik})=H) \quad (15)$$

for prespecified values of G and H for given i,j, and k.

Taking summations of equation (15) yields:

$$\sum\sum_{jk} P\left[T_{ijk} \cdot h(\ell_{ik}) \right] = \sum\sum_{jk} P_{G_{jk}} \cdot P_{H_{jk}} \quad (16)$$

Combining equations (14) and (16) yields the general form for the stochastic version of the nonlinear objective function.

B. Mathematical Solution to Model

1. Overview

This paper will only present the mathematical solution to the model with the deterministic nonlinear objective function. Solution of the stochastic version would be quite analogous. Below is a concise modified derivation based upon work of Gibbs (1978) and Mayhew (1981a) for the case of constrained demand, which is really the scenario in the United States for whose application the original models were not intended. The reader is referred to Gibbs (1978) and Mayhew (1981a) for a more detailed derivation of the solution to the supply constrained model.

Using standard optimization techniques for constrained functions with Lagrangian multipliers λ_i for $1 \le i \le m$, we form the Lagrangian:

$$H_i(T, \ell, \lambda) = U_i(T, \ell) + \lambda_i \left(S_i - \sum_k W_{ik}\, \ell_{ik} \right) . \quad (17)$$

To maximize the nonlinear preference objective function, it is necessary to solve the equations:

$$\frac{\partial H_i}{\partial T_{ijk}} = 0 \quad \text{for all i,j,k} \quad (18)$$

$$\frac{\partial H_i}{\partial \ell_{ik}} = 0 \quad \text{for all i and k} \quad (19)$$

$$\frac{\partial H_i}{\partial \lambda_i} = 0 \quad \text{for all i} \quad (20)$$

Equations (17), (5) and (18) yields:

$$\frac{dg(T_{ijk})}{dT_{ijk}} = \lambda_i \ell_{ik} - h(\ell_{ik}) . \quad (21)$$

Equations (17), (5), and (19) yields:

$$\frac{dh(\ell_{ik})}{d\ell_{ik}} - \lambda_i W_{ik} = 0 \quad (22)$$

Substituting equation (7) into equation (22) yields upon rearrangement:

$$\ell_{ik} = L_{ik}\lambda_i^{-\left(\frac{1}{\gamma_{ik}+1}\right)} \qquad (23)$$

Substituting equations (6) and (7) into equation (21) yields:

$$T_{ijk} = \phi_{ijk}(\mu_{ik})^{-\frac{1}{(\alpha_{ik}+1)}} \qquad (24)$$

where

$$\mu_{ik} = \frac{1}{\gamma_{ik}}\left[(\beta_{ik}+1)^{\left(\frac{\gamma_{ik}}{\gamma_{ik}+1}\right)}\lambda_i - 1\right]. \qquad (25)$$

Equations (23), (24), and (25) provide the solutions to the decision variables which minimize the nonlinear objective function given by equation (5) subject to the constraints given by equations (2), (3), and (4).

2. Parameter Estimation by Log-Linear Regression

The empirical elasticities for the lengths of stay ($\hat{b}^{\ell}_{ik}$) and the number of admissions ($\hat{b}^{W}_{ik}$) to the facilities from each origin i of each category k can be evaluated using log-linear regression as described below.

Taking logarithms of equation (2) with respect to the variables ℓ_{ik} and W_{ik} respectively, and extending all variables into dimension of time t yield the following two equations:

$$\log \overline{\ell_{ikt}} = \hat{a}^{\ell} + \hat{b}^{\ell}_{ik}(\log \overline{S_{it}}) + U_{it} \qquad (26)$$

$$\log \overline{W_{ikt}} = \hat{a}^{W} + \hat{b}^{W}_{ik}(\log \overline{S_{it}}) + Z_{it} \qquad (27)$$

In equations (26) and (27), U_{it} and Z_{it} are stochastic error terms; $\hat{a}^{\lambda}$ and $\hat{a}^{W}$ are constants; and $\overline{\ell_{ikt}}$, $\overline{W_{ikt}}$, and $\overline{S_{it}}$ are actual observations on average lengths of stay, generating factors, and total capacity respectively, as consumed by those originating from zone i in specialty k within planning horizon of duration t. The slope coefficients $\hat{b}^{\ell}_{ik}$ and $\hat{b}^{W}_{ik}$ of equations (26) and (27) respectively are precisely the empirical "elasticities" as defined previously.

C. *Algorithm for Parameter Estimation of Origin Constrained Model: Both Deterministic and Stochastic Versions*

1. Estimate T_{ijk} using origin constrained model as formulated by Mayhew (1980a).

2. Estimate empirical elasticities by log-linear regression.

3. Determine which parameters can be estimated or if probabilistic assumptions need be applied, i.e. select deterministic or stochastic version.

4. Using these parameter values predict ℓ_{ik} and T_{ijk} solving equations analogous to equations (23) and (24) for either version. It may be useful to perform sensitivity analysis for predicting ℓ_{ik} and T_{ijk} under varying combinations of W_{ik} and capacity S_i.

III. Some Results of Simulation

The goodness of the parameters estimated are determined by performing simulations on actual input data and comparing these predicted T_{ijk} flows with the actual flows. Several standard statistical tests can be used to determine the best set of parameter values. In Table 1 below, results are presented using the R^2 statistic, which as usual gives the proportion of explained variance.

In Table 1, some results are presented for simulation runs of using the deterministic DRAMOS model in its destination constrained form with data representing hospital discharges in multi-category specialties for the State of Massachusetts in 1978. Three parameter estimation techniques were used for model calibration as shown in Table 1: slope=1.0 calibration, maximum likelihood, and maximum R^2. These simulations show that the maximum R^2 calibration method yielded the highest R^2 values and maximum likelihood generally yielded the lowest. In Table 1, the parameter β is the calibration coefficient value of multi-categorical extension for model of Mayhew (1980a), which yielded the corresponding R^2 value.

Table 1: Calibration of DRAMOS using 1978 inpatient discharge data from Massachusetts

Category of patient care	Number of patient discharges	SLOPE=1.0 CALIBRATION β	R^2
Total (all patients)	851760	.1600	.8407
Medical-Surgical	658942	.1600	.8395
Obstetric-Maternity	88192	.1900	.8920
Pediatric	84391	.1500	.7678
Psychiatric	20182	.1900	.8635

Category of patient care	MAXIMUM LIKELIHOOD CALIBRATION β	R^2
Total (all patients)	.1175	.8577
Medical-Surgical	.1151	.7998
Obstetric-Maternity	.1408	.8588
Pediatric	.1112	.7245
Psychiatric	.1392	.8664

Category of patient care	MAXIMUM R^2 β	R^2
Total (all patients)	.6100	.8803
Medical-Surgical	.6100	.8759
Obstetric-Maternity	.5100	.9218
Pediatric	.4100	.8350
Psychiatric	.8100	.9053

IV. Conclusions and Future Directions

This research extends a simulation model for predicting multi-categorical flows within large-scale market oriented systems. Both deterministic and stochastic simulation versions have been presented with some results for the former version.

The future directions include more extensive simulation runs for the deterministic version. Also the mathematical solution to the stochastic version of the model should be completed in order to provide some results of stochastic simulation.

V. Bibliography

R.J. Gibbs: A Disaggregated Health Care Resource Allocation Model, RM-78-1 (IIASA, Laxenburg, Austria, 1978a).

R.J. Gibbs: The IIASA Health Care Resource Allocation Sub-Model: Mark 1, RR-78-8 (IIASA, Laxenburg, Austria, 1978b).

D. Hughes and A. Wierzbicki: DRAM: A Model of Health Care Resource Allocation, RR-80-23 (IIASA, Laxenburg, Austria, 1980).

G. Leonardi, A Multiactivity Location Model with Accessibility and Cogestion Sensitive Demand, WP-80-79 (IIASA, Laxenburg, Austria, 1980).

G. Leonardi, The Use of Random Utility Theory in Building Location-Allocation Models, WP-81-28 (IIASA, Laxenburg, Austria, 1981).

L. Mayhew and A. Taket, RAMOS: A Model of Health Care Resource Allocation in Space, WP-80-125 (IIASA, Laxenburg, Austria, 1980a).

L.D. Mayhew, The Regional Planning of Health Care Services: RAMOS and RAMOS^{-1}, WP-80-166 (IIASA, Laxenburg, Austria, 1980b).

L.D. Mayhew, DRAMOS: A Multi-Category Spacial Resource Allocation Model for Health Service Management and Planning, WP-81-39 (IIASA, Laxenburg, Austria, 1981a).

L.D. Mayhew and A. Taket, RAMOS: A Model Validation and Sensitivity Analysis, WP-81-100 (IIASA, Laxenburg, Austria, 1981b).

R.S. Segall, P.G. Abrahamson, E.J. Rising, D.C. Sonderman, Models of Area Wide Medical Care Delivery, Event WA21.1, ORSA/TIMS Joint National Meeting, San Diego, CA, October 1982.

R.S. Segall, E.J. Rising, P.G. Abrahamson and D.C. Sonderman, Mathematical Modeling of Patient Travel and Use of Hospitals in Massachusetts, Event MB19.4, ORSA/TIMS Joint National Meeting, Orlando, FL, November 1983.

E.J. Rising and L.D. Mayhew, The Spatial Allocation of Medical Care Resources in Massachusetts: An Application of RAMOS, WP-83-38 (IIASA, Laxenburg, Austria, 1983).

E.J. Rising and R.S. Segall, A Model of Patient Use of Hospitals in Massachusetts, Proceedings of Hospital Management Systems Society Annual Conference, San Francisco, CA, February 1984a.

E.J. Rising and R.S. Segall, The Calibration of an Origin Constrained Gravity Model to Predict Patient Flow, Proceedings of the Third International Conference on Systems Science in Health Care, Munich, Germany, July 16-20, 1984b.

R.S. Segall, Models of Area Wide Medical Care Delivery, Ph.D. Thesis, University of Massachusetts at Amherst, MA, 1984.

R.S. Segall and E.J. Rising, Some Strategies for Using an Origin Constrained Gravity Model for Decision Making of Hospital Capacity, Proceedings of Northeast American Institute for Decision Sciences, Williamsburg, VA, March, 1986.

R.S. Segall and E.J. Rising, A Model for Forecasting Hospital Bed Requirements, Event WC14.2, Seventh International Symposium on Forecasting, Boston, MA, May 1987a.

R.S. Segall, Mathematical Programming for the Optimization of Objectives of Health Care Systems, Contributed Paper presented at Northeastern Section of the Mathematical Association of America, New London, CT, June 13, 1987b.

R.S. Segall and E.J. Rising, The Application of Computer Technology for the Capacity Planning and Marketing of Hospitals, Proceedings of the Seventh International Business School Computer Users Group, Flint, MI, July 1987c.

R.S. Segall, Some Nonlinear Optimization Models for Planning of Large Scale Systems, Event MD37.5, ORSA/TIMS Joint National Meeting, Washington, DC, April 1988a.

R.S. Segall, Mathematical Modelling for the Capacity Planning of Market Oriented Systems: with an application to real health data, Applied Mathematical Modelling, v.12, August 1988b.

D.D. Venedictov and E.N. Shigan, The IIASA Health Care System Model, Paper presented at IIASA Conference on Modeling Health Care Systems, (IIASA, Laxenburg, Austria; November 1977).

Address Correspondence to:

Dr. Richard S. Segall
Department of Mathematics
University of Lowell
Lowell, MA 01854

A Monte Carlo Assessment of Cross-validation and the C_p Criterion for Model Selection in Multiple Linear Regression

Robert M. Boudreau, Virginia Commonwealth University

1. Introduction

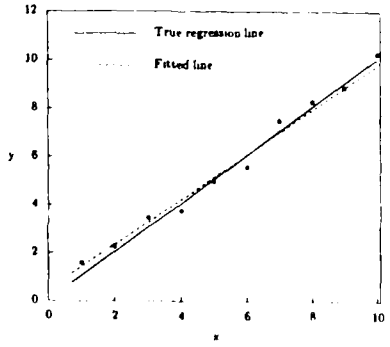
Consider the situation of fitting a multiple linear regression to a set of data. The data consists of n observations on some response variable, together with corresponding observations on p predictor variables. The ultimate use of the fitted model will be as a prediction equation. The current data is to be used to assess and select the "best" subset of the predictor variables, and to provide estimates of the regression coefficients for these variables. "Best" might be defined in terms of smallest mean squared error of prediction (MSEP), or smallest mean absolute deviation of prediction (MADP). Keep in mind that the "true" model is not necessarily the "best" model for prediction purposes (p 248 Montgomery and Peck, 1982). The goal here is different than the model building of a researcher/scientist seeking to explain and understand the relationships between the predictors and the response variable. There the model sought should contain the correct or complete set of regressors, with accurate estimates of the coefficients. The distinction is useful because the different criteria used for model selection are usually motivated by and align with one or the other of these intended uses of the fitted model.

2. Unconditional vs Conditional MSEP

No fitted model ever contains the exact values of the parameters, since these are estimated from the data. The parameter estimates are unbiased if we overfit, and biased as estimates of the true values of the parameters if we underfit (p 247 Montgomery and Peck, 1982). This unbiasedness or biasedness refers to the average behavior of parameter estimates averaged over many researchers, studies, or data sets. Similarly, a prediction equation is unbiased or biased on average for a response to be predicted depending on whether we've overfit or underfit.

In terms of squared error of prediction, there is an average, or unconditional mean squared error of prediction (MSEP) in using a particular subset of variables. Mallows C_p (Mallows, 1973) (p 252 Montgomery and Peck, 1982) is often motivated as an estimate of MSEP when the predictors are fixed.

A natural, related question arises. Since the parameter estimates are not exact, and are conditional on the training sets, which estimated model best predicts new responses? The new responses adhere to the "true" model, while our fitted model differs conditional on the training data (see below).



This seems more realistic in the sense that it will be our parameter estimates that will be used. One would like to pick the fitted model with the smallest conditional mean squared error of prediction (CMSEP).

The results of sections 3 and 4 show that the C_p criterion (fixed predictors) and cross-validation (random predictors) are uncorrelated with the CMSEP's for the fitted models. These criteria therefore cannot be interpreted as estimates of the CMSEPs for a particular data set. I point out here that these results are for multiple linear regression with normal errors. Cross-validation has wider application. It is an open question whether cross-validation is uncorrelated with CMSEP for more general regression functions, errors, and approximating prediction functions (Efron, 1983).

3. Fixed Predictors: C_p

Let the current training data for assessing and selecting a prediction equation satisfy the following:

$$\underset{n\times 1}{y} = \underset{n\times p}{X}\ \underset{p\times 1}{\beta} + \underset{n\times 1}{\epsilon} \qquad (1)$$

where y is a vector of responses, X is a fixed full rank matrix of predictors, and the elements of ϵ are iid $N(0,\sigma^2)$. Predictors for a submodel proceeds as follows. Select a subset of k variables, form X_k by including only columns of X from these variables, and fit a prediction equation by least squares:

$$\hat{y}_k = X_k\hat{\beta}_k = P_k y \qquad (2)$$

where

$$\beta_k = (X_k' X_k)^{-1} X_k' y$$

$$P_k = X_k (X_k' X_k)^{-1} X_k' y$$

$$E_\epsilon[\hat{\beta}_k] \neq \beta_k$$

Paralleling Efron (1986), consider predicting new studies to be conducted with the same design matrix X as the training data, with responses y_0.

$$y_0 = X\beta + \epsilon_0 \qquad (3)$$

Then the CMSEP for the training set (1) in predicting new responses is given by

$$\begin{aligned} CMSEP_k &= E_{\epsilon_0}\left[\|y_0 - \hat{y}_k\|^2\right] \\ &= \tfrac{1}{n}\|X\beta - X_k\hat{\beta}_k\|^2 + \sigma^2 \end{aligned} \qquad (4)$$

where $\hat{\beta}_k$ is fixed. Then averaging over training data sets (1) yields the average CMSEP

$$MSEP_k = E_\epsilon[CMSEP_k] = \tfrac{1}{n}\|X\beta - P_k X\beta\|^2 + \tfrac{n+k}{n}\sigma^2 \qquad (5)$$

$MSEP_k$ is the mean squared prediction error for the subset of k variables if all researchers use these variables.

As pointed out by Efron (1986), the statistic

$$C_k = \frac{1}{n} ||y - X_k\hat{\beta}_k||^2 + 2k\hat{\sigma}^2 \quad (6)$$

is equivalent to Mallow' C_p for the k variables,

where $\hat{\sigma}^2 = \frac{1}{n-p} ||y - X\hat{\beta}||^2$ (using all predictors)

$$= \frac{1}{n-p} ||y - Py||^2;$$

and $P = X(X'X)^{-1}X^1$.
It is easy to show that

$$E[C_k] = E[CMSEP_k] = MSEP_k$$

Thus C_k unbiasedly estimates $MSEP_k = E[CMSEP_k]$.

Asymptotically Stone (1977) showed that the C_p, AIC (Aikaike, 1973) and Cross-validation (CV) (Stone, 1974) are equivalent. Nishii (1984) showed that asymptotically the C_p and CV don't underfit, but they do overfit with non-zero probabilities. Li (1987) showed that the C_p and CV asymptotically yield the best CMSEP.

For smaller n, are C_p and CMSEP related? First note that for training data (1), C_k of (6) expands as

$$\begin{aligned} C_k &= \tfrac{1}{n} y'(I - P_k)y + \frac{2k}{n(n-p)}y'(I-P)y \\ &= \tfrac{1}{n}\beta'X'(I-P_k)X\beta + \tfrac{1}{n}\epsilon'(I-P_k)\epsilon \\ &+ \tfrac{2}{n}\beta'X'(I-P_k)\epsilon + \frac{2k}{n(n-p)}\epsilon'(I-P)\epsilon \end{aligned} \quad (8)$$

The random part of C_k involves a sum of two quadratic forms and a linear form in ϵ. Similarly, $CMSEP_k$ of (4) expands as

$$\begin{aligned} CMSEP_k &= \tfrac{1}{n}||X\beta - P_k(X\beta + \epsilon)||^2 + \sigma^2 \\ &\vdots \\ &= \tfrac{1}{n}\beta'X'(I-P_k)X\beta + \sigma^2 + \tfrac{1}{n}\epsilon' P_k\epsilon \end{aligned} \quad (9)$$

Observe that

$$P_k(I-P_k) = P_k - P_k = 0$$

$$P_k(I-P) = P_k - P_k = 0$$

Next noting conditions for independence of quadratic and linear forms in normal variables (Rao, 1973), we have that C_k and $CMSEP_k$ are in fact uncorrelated (independent). C_k must exclusively be considered an estimate of $MSEP_k = E[CMSEP_k]$, not $CMSEP_k$. Mallows C_p recommends a set of variables to the wider community. C_k is an estimate of the $MSEP_k$. Whether your $CMSEP_k$ for those variables is higher or lower than average ($MSEP_k$) you don't know.

4. Random Predictors: Cross-validation.

Next consider developing a linear least squares prediction equation when the matrix X of predictors in model (1) is random. This is usually the case in practice. When parametric model (1) is true, then various criteria derived assuming this are available and appropriate. S_p(reviewed by Hocking, 1976 and Thompson, 1978) is an estimator of the unconditional mean squared error averaged over multivariate normal predictors and response. It is directly analogous to C_p for fixed predictors. Also available are an AIC assuming multivariate normality (Aikaike, 1973) and an AIC without assuming a particular covariance structure for X (comment by Aikaike on Rao, 1987).

For a more general (possibly non-linear) unknown true regression function of y on X, assessing the performance of a linear prediction equation cannot be assessed by the above criteria. Various non-parametric estimators of the prediction error have been proposed, including the jackknife, the bootstrap (Efron, 1979 and Efron and Gong, 1983), and cross-validation (Allen, 1971; Stone, 1974; Geisser, 1975). This paper investigates a property of cross-validation as a method of assessing a prediction equation when the multivariate normal linear model holds.

The version of cross-validation considered here is the leave-one-out at a time method. One observation and the corresponding set of predictors is omitted. The remaining data, consisting of n−1 observations on responses and predictor variables, is used to fit a least squares linear prediction equation (or any general prediction equation). The regression parameter estimates thus obtained are used to predict the omitted response. The squared difference is recorded. The process is repeated, omitting each response/predictor pair temporarily until each response has been predicted using the remaining n−1. The process mimicks the process of predicting new observations. The apparent error rate

$$\tfrac{1}{n} ||y - X_k\hat{\beta}_k||^2 \quad (10)$$

which is closely related to the usual regression mean residual sum of squares, is known to considerably underestimate the actual prediction squared error in the random regressor case (Efron, 1986). This stems from two problems. The data is used to predict itself, so tends to be optimistic. Further, because future predictors are random, the training set of predictors, X, doesn't represent the full variation of future observations unless n is very large. This underestimation is one of the basic motivations for bias correction such as the jackknife and the bootstrap.

Let $X_{k(i)}$ be the matrix composed of columns of X corresponding to variables in some subset of k of the total p predictor variables, but with the ith row deleted. Similarly, let $y_{(i)}$ be the vector of responses with the ith response omitted.

Then fitting

$$y_{(i)} = X_{k(i)}\beta_k + \epsilon$$

yields

$$\hat{\beta}_{k(i)} = (X'_{k(i)}X_k)^{-1}X'_{k(i)}y_{(i)} \quad (11)$$

The least square predictor of the omitted response, y_i, using the above estimated parameters from the remaining n−1 observations, is the product of the observations on the k predictors for the i-th response times the corresponding regression coefficients (11). Denote the predictor by $\hat{y}_{i(-i)}$.

The average of the squared prediction errors, called PRESS by Allen (1973), and called CVAE by Rao (1987) abbreviating Stone's (1974) cross-validatory assessment error, will be denoted here as

$$CV_k = \tfrac{1}{n}\Sigma(y_i - \hat{y}_{i(-i)})^2 \quad (12)$$

As was pointed out by A.P. Dawid in his comment on Stone's 1974 paper on cross-validation, and also by Efron (1986), CV_k is an unbiased estimate of the unconditional mean squared error of prediction when selecting a training set of random predictors of size n−1 to develop a prediction equation, then using that equation, to predict new observations. The unconditional mean

squared error of prediction when predictors are random will be denoted $MSEP_+$.

In the present context, the subset of the variables chosen is the subset that minimizes CV_k of (12). Having chosen a subset, the full training set on all n observations is then used to estimate the parameters to be used for prediction. Both Dawid and Efron, in the papers just referred to, point out that CV is based on training set of size $n-1$. Consequently, CV underestimates, or is biased downward, for the unconditional $MSEP_+$ for a given subset because the final parameter estimates will be based on the entire training set, which has size n. Dawid's exact expression in the multivariate normal case, and Efron's (1986) simulations for a linear fit to a quadratic (true but unknown) regression function indicate that the bias is small, as might be expected.

The same question as in section 3 arises here concerning conditional mean squared error of prediction ($CMSEP_+$). The prediction equations use estimated parameters. Which conditional prediction equation, conditional on the current training set, yields the smallest $CMSEP_+$? Thompson (1978) motivates S_p and PRESS (same as CV) as estimates of $CMSEP_+$. Picard and Cook (1984), Efron (1983, 1986) and Rao (1987) also view CV as an estimate of $CMSEP_+$. In what follows evidence is given to suggest that a CV assessment obtained from a given data set for a subset of variables is uncorrelated with the $CMSEP_+$ for that data set. This runs counter to intuition since CV actually simulates the process of prediction by withholding independent observations to be used for prediction. As in section 3, the results that follow are restricted to multivariate normal predictors and response. The behavior of CV in the general case is ultimately of interest because that is the more appropriate situation to use cross-validation. The multivariate normal case is a first step.

CV_k of (11) expands as (eg. p 430, Montgomery and Peck, 1982)

$$\begin{aligned} CV_k &= \tfrac{1}{n}\Sigma(y_i-\hat{y}_{i(-i)})^2 \\ &= \tfrac{1}{n}\, y'(I-P_k)(I-D_k)^{-2}(I-P_k)y \\ &= \tfrac{1}{n}(X\beta+\epsilon)'Q_k(X\beta+\epsilon) \\ &= \tfrac{1}{n}(\beta'X'Q_kX\beta + 2\beta'XQ_k\epsilon + \epsilon'Q_k\epsilon) \end{aligned} \quad (13)$$

where

$$D_k = \text{diag}\,(I-P_k)$$

$$P_k = X_k(X_k'\,X_k)^{-1}X_k'$$

$$Q_k = (I-P_k)(I-D_k)^{-2}(I-P_k)$$

Like C_k, CV_k is also a sum of a quadratic and linear form in ϵ. The difference is that X is random, so that the coefficient matrices of the quadratic form and the linear form are random. They are fixed exactly like C_k, conditional on X.

Let training set of size n be given by

$$\underset{n\times 1}{y} = \underset{n\times p}{X}\ \underset{p\times 1}{\beta} + \underset{n\times 1}{\epsilon} \quad (14)$$

where X is multivariate normal $N(\mu,\Sigma)$
ϵ is a vector of iid $N(0,\sigma^2)$ independent of X

Fit a linear regression to a subset of k variables as in (2), yielding coefficients $\hat{\beta}_k$ (size $k\times 1$). Form $p\times 1$ vector $\hat{\beta}_{k+}$ by putting elements of $\hat{\beta}_k$ in the appropriate positions corresponding to the k variables in the subset, then setting the remaining values to 0.

Given a new observation

$$y_0 = x_0'\beta + \epsilon \quad (15)$$

the predicted value using the k subset is

$$\hat{y}_{0k} = x_0'\hat{\beta}_{k+} \quad (16)$$

Averaged over new observations, the $CMSEP_+$ for the k subset based on the training data (14) is

$$\begin{aligned} CMSEP_{k+} &= E_{x_0,\epsilon_0}\left[(y_0-\hat{y}_{0k})^2|X,\epsilon\right] \\ &= E_{x_0,\epsilon_0}\left[x_0'\beta+\epsilon_0-x_0'\hat{\beta}_{k+})^2|X,\epsilon\right] \\ &= (\hat{\beta}_{k+}-\beta)(\Sigma+\mu\mu')(\hat{\beta}_{k+}-\beta) \\ &\ \ \vdots \\ &= C + M_1\epsilon + \epsilon'M_2\epsilon \end{aligned} \quad (17)$$

where M_1 and M_2 depend only on X.

Conditional on training set X, it can be shown that:

$$Q_kM_1 = 0$$

$$Q_kM_2 = 0$$

Thus CV_k (13) and $CMSEP_{k+}$ (17) are conditionally (locally) uncorrelated for every set of predictors (14). In general, conditionally uncorrelated does not imply unconditionally uncorrelated. Based on simulations, however, there is strong evidence that CV_k and $CMSEP_{k+}$ are unconditionally uncorrelated.

The following are the results of two of many simulation results the author has run. The interpretation is the same in every simulation that has been tried. The simulations were performed in SAS PROC MATRIX, with 1000 Monte Carlo iterations per experiment. In the first two experiments below, training sets of sample size $n=30$ with 5 multivariate normal predictors with means equal to 0 (wlog), and a dependent response were generated. The last two variables are superfluous by design.

Experiment #1

$$y_i = .1 + X_1 + .5X_2 + .25X_3 + 0X_4 + 0X_5 + \epsilon_i \quad (i = 1,\ldots,30)$$

where

$$\text{Cov}(X_1,\ldots,X_5) = \begin{bmatrix} 1 & .2 & .2 & 0 & 0 \\ .2 & 1 & .2 & 0 & 0 \\ .2 & .2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and $\epsilon_i \sim N(0, 0.25)$.

There are 2^6-1 possible subsets of the variables (constant included). Presented here are the results for 6 of these as typical cases.

In a hierarchical fashion:

Fitting a constant yields CV_0
Fitting a constant plus X_1 yields CV_1

Fitting a constant plus X_1, X_2 yields CV_2

.

.

.

Fitting a constant plus X_1, X_2,..., X_5 yields CV_5.

Each of the 6 fitted models has a corresponding $CMSEP_{n+}$. After the 1000 iterations, the correlations between the CV's and $CMSEP_+$'s are given below.

PEARSON CORRELATION COEFFICIENTS / PROB > |R| UNDER H0 RHO=0 / N = 1000

	CV0	CV1	CV2	CV3	CV4	CV5
CMSEP0	-0.02447	0.01704	0.03739	0.03798	0.04474	0.03051
	0.4395	0.5905	0.2375	0.2301	0.1574	0.3351
CMSEP1	-0.04434	0.01228	-0.00702	0.00076	0.00097	-0.00632
	0.1612	0.6987	0.8025	0.9808	0.9756	0.8417
CMSEP2	-0.05515	-0.04619	-0.02625	-0.01572	-0.02532	-0.03107
	0.0813	0.1444	0.4078	0.6194	0.4237	0.3264
CMSEP3	-0.03039	0.00386	0.05645	-0.01284	-0.01989	-0.02385
	0.3370	0.9029	0.0744	0.6850	0.5299	0.4513
CMSEP4	-0.01983	0.03285	0.14788	0.11436	0.00816	-0.00320
	0.5310	0.3113	0.0001	0.0003	0.7967	0.9194
CMSEP5	0.00482	0.03856	0.18518	0.16546	0.07112	-0.01128
	0.8991	0.2231	0.0001	0.0001	0.0245	0.7217

The striking feature is that the estimated correlations between CV_k and $CMSEP_{k+}$ are all very small, with p-values all larger than 0.4. The same is true for the remaining 57 possible subsets (i.e. none having correlations significantly different from 0). The null hypothesis that CV and $CMSEP_+$ are uncorrelated is accepted (statistically speaking) .

Experiment # 2

$$y_i = .25 + .5X_1 + .1X_2 + X_3 + 0X_4 + 0X_5 + \epsilon_i \quad (i = 1,...,30)$$

where

$$Cov(X_1,.....,X_5) = \begin{bmatrix} 1 & .5 & .5 & .2 & 1 \\ .5 & 1 & .5 & .2 & 1 \\ .5 & .5 & 1 & .2 & 1 \\ .2 & .2 & .2 & 1 & 1 \\ .1 & .1 & .1 & .1 & 1 \end{bmatrix}$$

and $\epsilon_i \sim N(0, 0.25)$.

PEARSON CORRELATION COEFFICIENTS / PROB > |R| UNDER H0 RHO=0 / N = 1000

	CV0	CV1	CV2	CV3	CV4	CV5
CMSEP0	0.03508	0.03835	0.04710	0.04508	0.04302	0.02485
	0.2677	0.2256	0.1366	0.1543	0.1740	0.4325
CMSEP1	0.06244	0.01169	0.01519	-0.00308	-0.00896	-0.01265
	0.0484	0.7128	0.6315	0.9225	0.7772	0.6894
CMSEP2	0.03528	0.00688	-0.03175	-0.01825	-0.02749	-0.02486
	0.2649	0.8281	0.3159	0.5644	0.3852	0.4472
CMSEP3	-0.02644	-0.02915	-0.03424	-0.01284	-0.01989	-0.02385
	0.4036	0.3571	0.2794	0.6850	0.5299	0.4513
CMSEP4	-0.01057	0.00862	0.02354	0.11436	0.00816	-0.00320
	0.7385	0.7855	0.4571	0.0003	0.7967	0.9194
CMSEP5	0.00487	0.01442	0.04258	0.16546	0.07112	-0.01128
	0.8779	0.6488	0.1785	0.0001	0.0245	0.7217

As in experiment #1, and in every case tried, the correlations between CV and $CMSEP_+$ are not significantly different from 0.

Experiment # 3 (Efron, 1986)

$$y_i = X_i + 0.01\,X_i^2 + \epsilon_i \quad (i = 1, ..., 20)$$

$$X_i \sim N(0, 10^2) \ , \ \epsilon_i \sim N(0, 1)$$

Fitting a simple linear regression to this quadratic (non-linear) data

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

has a corresponding cross-validatory assessment CV. The following, reproduced from Efron (1986), gives the first 10 of 20 Monte Carlo trials, plus a summary of all 20 trials. Err_+ means the same as $CMSEP_+$ in this paper. The fundamental difference in experiment #3 is that Efron's true simulated regression is quadratic while the model fit is a linear one. He also compares the bootstrap estimate of $CMSEP_+$.

	CV (7.10)	$\widehat{Err}^{boot}$ (B = 400)	Err. (7.14)
	3.36	3.19	3.42
	3.84	2.97	3.21
	4.73	4.48	2.79
	4.09	3.66	3.42
	2.84	2.67	3.11
	4.80	4.22	2.86
	2.31	2.17	3.46
	3.01	2.54	2.72
	3.67	3.16	4.91
	4.13	3.84	3.72
20 Trials (AVE):	3.26	2.97	3.39
20 Trials (SD):	(1.15)	(.98)	(.54)

Notice that the CV is nearly unbiased for average $CMSEP_+$, while the bootstrap is biased downward considerably more. On the other hand, the bootstrap is closer to $CMSEP_+$ in a mean squared error sense than is CV (see Efron 1986 for details). Efron gives evidence that the bootstrap is a "somewhat" better estimate of $CMSEP_+$ than is CV, although not strongly so judging by his simulation.

The point made in this paper is that computing the correlation between CV and $CMSEP_+$ in Efron's simulation yields $r = -.11$ and p-value $= .72$. There is again evidence that CV and $CMSEP_+$ are uncorrelated, this time when the true regression is non-linear. This author maintains that it is awkward to interpret one random variable (CV) as an estimate of another random variable ($CMSEP_+$) when the two are uncorrelated. The mean squared error of their difference comes solely from their respective variances and the squared difference in their expected values. One doesn't track the other in any discernible way if they're uncorrelated. Cp and CV must be seen as estimates of the unconditional mean squared error of prediction of a subset of variables. .

Reference.

Akaike, H. (1973). "Information theory and an extension of the maximum likelihood principle." *2nd International Symposium on Information Theory* 267-281. (B.N. Petrov and F. Czáki, eds). Akademiai Kiadó, Budapest.

Allen, D. (1971). The prediction sum of squares as a criterion for selecting prediction variables. Univ. of Kentucky, Dept. of Statistics, Technical Report, No. 23.

Efron, B. (1979), "Bootstrap Methods: Another Look at the Jackknife." *Annals of Statistics*, 7, 1-26.

_____ (1983), "Estimating the Error Rate of a Prediction Rule: Improvements on Cross-Validation," *Journal of the American Statistical Association*, 78, 316-331.

_____ (1986), "How Biased is the Apparent Error Rate of a Prediction Rule?", *Journal of the American Statistical Association*, 81, 461-470.

Efron, B., and Gong, G. (1983), "A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation," *The American Statistician*, 37, 36-48.

Geisser, S. (1975), "The Predictive Sample Reuse Method with Applications," *Journal of the American Statistical Association*, 70, 320-328.

Hocking, R.R. (1976), "The analysis and selection of variables in linear regression," *Biometrics*, 32, 1-49.

Li, K. (1987), "Asymptotic Optimality for C_p, C_L, Cross-Validation and Generalized Cross-Validation: Discrete Index Set," *The Annals of Statistics*, 15, 958-975.

Mallows, C.L. (1973), "Some comments on C_p," *Technometrics*, 15, 661-675.

Montgomery, D. and Peck, E. (1982), *Introduction to Linear Regression Analysis*, Wiley, New York.

Nishii, R. (1984), "Asymptotic Properties of Criteria for Selection of Variables in Multiple Regression," *The Annals of Statistics*, 12, 758-765.

Picard, R.R., Cook, R.D. (1984), "Cross-Validation of Regression Models," *J. Amer. Statist. Ass.*, 79, 575-583.

Rao, C.R. (1973), Linear Statistical Inference and Its Applications, 2nd ed. Wiley, Ne York.

_____ (1987), "Prediction of Future Observations in Growth Curve Models," *Statistical Science*, 2, 434-471.

Stone, M. (1974), "Cross-Validiatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society*, Ser. B, 36, 111-147.

_____ (1977), "An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion," *Journal of the Royal Statistics Society*, Ser. B, 39, 44-47.

Thompson, M.L. (1978), "Selection of variables in multiple regression: Part II. Chosen procedures, computations and examples," *Int. Statist. Rev.*, 46, 129-146.

IT'S TIME TO STOP

by
Hubert Lilliefors
The George Washington University

ABSTRACT: Simulations are frequently used to estimate certain characteristics of a distribution. A question that arises is how large a sample should we use? We consider specifically the estimation of population quantiles. The procedure presented here relies on the large sample normality of sample quantiles. This requires an estimate of the density function evaluated at the quantile. An apparently new estimator is used and is compared to the Siddiqui estimator. Simulation results are used to compare the estimators and also to compare several stopping procedures.

1. **INTRODUCTION:** Simulations are frequently used to estimate certain characteristics of a distribution such as the mean or the median. In the case explicitly considered in this paper the estimated characteristics are the 90th, 95th and 99th quantiles which are appropriate when generating critical values for some test statistic. The test statistic is generated independently a large number of times and then the sample quantile is used as an estimate of the population quantile.

The question is : how large is large? When can the simulation be stopped? Should there be 500 repetitions? or 5000 repetitions? or 500,000 repetitions? One approach is to generate confidence intervals sequentially until a prescribed fixed width is obtained. This is discussed in Law and Kelton (1982) for estimation of the mean of a distribution (additional references are also given). Recently Dallal and Wilkinson (1986) used this type of procedure for quantile estimation. They started with a sample size of 50000 and computed a 95% confidence interval for the 99th quantile. If the width of the interval was less than some prescribed width (they used .001) they stopped. Otherwise they added another 50000 to the sample and tried again. This continued until either their condition was satisfied or they reached an upper limit on the sample size.

In this paper an alternative (large sample) procedure is presented for determining when to stop a simulation when the purpose of the simulation is to estimate a population quantile. This procedure uses an (apparently new) estimator for the value of a density function evaluated at a particular quantile. The method compares favorably to the Dallal & Wilkinson procedure in a rather limited simulation. The same method might be used when estimating other characteristics of a distribution.

2. **THE NEW PROCEDURE:** The alternative procedure makes use of the well known asymptotic (normal) distribution of sample quantiles (see for example David (1970)). If we require a 95% probability that the sample quantile is within a distance B of the population quantile, then the sample size required is given by:

$$N = p(1-p)1.96^2/(B^2 f(x_p)^2), \qquad (2.1)$$

where x_p is the pth population quantile, and an estimate is needed for the density function evaluated at the population quantile.

Two density estimators were used, the Siddiqui (1960) estimator and a new least squares estimator. These are discussed in the next section.

The procedure has two stages. In the first stage a preliminary sample is drawn to provide an estimate of the density function evaluated at the quantile of interest. Using this estimate, and (2.1) an estimate is obtained of the required total sample size, and hence the size of the additional sample needed. The second sample is drawn and the sample quantile is determined from the combined samples to provide an estimate of the population quantile.

A (three stage) variation on this procedure was also tried in which, after the second sample is drawn, we again estimate the density function and if a larger sample is determined to be necessary we draw another sample.

3. **DENSITY ESTIMATORS:** In order to use (2.1) to determine when to stop the simulation, an estimate is required for the reciprocal of the density function evaluated at the quantile of interest. There has been a great deal of work on estimating density functions. Silverman (1986) provides a good general description of the basic techniques. Any of these techniques might be used to obtain an estimate of the density function and then evaluate this estimated density function at the estimate of the quantile.

A rather clever procedure which avoids using two estimates was suggested by Siddiqui (1960), further developed by Bloch and Gastwirth (1968) and by Bofinger (1975). A second estimator, which uses a least squares calculation (see also Eldessouky (1985)) was suggested by the form of the Siddiqui estimator. These are described below.

a) **SIDDIQUI ESTIMATOR:** We follow the development in Bofinger (1975) with slight changes in notation.

The Siddiqui estimator for $1/f(x_p)$ is

$$T_{(x)} = (X_{(m)} - X_{(n)})/(Y_{(m)} - Y_{(n)}) \qquad (3.1)$$

where N is the sample size

$$m = [N(p+d_m)]+1$$

$$n = [N(p+d_n)]+1$$

For the case $d_m = d_n = d$, the equation for $T_{(x)}$ becomes:

$$T_{(x)} = (X_{(m)} - X_{(n)})/2d \qquad (3.2)$$

and Bofinger shows that asymptotically the optimum (m.s.e. sense) value for d is

$$d = C/N^{.2} \qquad (3.3)$$

where

$$C = 1.5^{.2}\{2[F'(x_p)]^4[3(F''(x_p)/F'(x_p))^2 - (F'''(x_p)/F'(x_p))^2/9]\}^{-.2} \qquad (3.4)$$

The table below shows the values for C calculated from (3.4). The values of these (optimal) values for C do not change a great deal as we go from the heavy tailed exponential distribution to the light tailed Weibull (with shape parameter =4) distribution.

TABLE 1 OPTIMAL VALUE FOR C

	QUANTILE			
DISTRIBUTION	.50	.90	.95	.99
Exponential	.5880	.1623	.0932	.0257
Normal	.6447	.1876	.1043	.0277
Weibull(=4)		.2286	.1349	.0415

Intuitively it is fairly clear why the Siddiqui estimator works. If for x in some region $x_{(n)} < x < x_{(m)}$ (where $x_{(n)}$ and $x_{(m)}$ are the order statistics defined above) the cumulative distribution function, F(x), is approximately linear, then the slope of that line is the density function, f(x), and if x_p (the p^{th} quantile) lies within that interval $f(x_p)$ will equal that slope. Thus we want to take an interval narrow enough so that the approximate linearity holds.

b) **LEAST SQUARES ESTIMATORS** We note that there will be many data points between $x_{(n)}$ and $x_{(m)}$, but that for the Siddiqui estimator we simply connect the two extreme points to get an approximation to F(x) in that interval (and then use the slope of that line as the estimate for the density function f(x)).

Two possibilities occur immediately:

(i) The straight line approximation is good, but why not use all the data? Using the same notation as in (2.1) above, we use the end points and 18 (evenly spaced) additional points between X(n) and X(m) for a total of 20 points and then use the ordinary least squares procedure to fit a straight line to the data. This seemed to work about as well as the Siddiqui procedure, but was no improvement. Using all the points between $x_{(n)}$ and $x_{(m)}$ gave almost exactly the same results but required considerably more time for the computations. (A weighted least squares might have given an improvement but was not tried.)

(ii) If a straight line approximation to F(x) works well, then why not try a quadratic approximation using the ordinary least squares fit to the 20 points as described above. This will give

$$F(x) = b_0 + b_1x + b_2x^2$$

and from this $f(x) = b_1 + 2b_2x$

and to estimate $f(x_p)$, the density function evaluated at the p^{th} quantile, the p^{th} sample quantile as an estimate of the p^{th} population quantile.

This seems to have two advantages over the Siddiqui estimator:

(a) It is more accurate based on a rather limited simulation.

(b) It is less sensitive to the choice of d (or equivalently to the choice of the endpoints of the interval - but see the discussion below of the simulation results with the exponential distribution.)

For the interval of x values, we use an interval of the same form as that for the Siddiqui estimator (see (3.2) and (3.3)) and try different values for C (see (3.3)).

4. **SIMULATION TO DETERMINE THE VALUE FOR C:** A simulation was used to determine the sensitivity of the Siddiqui estimator to the choice of C and to determine which value of C to use with the least squares estimator. The simulation was performed for each of three distributions ranging from light tail (Weibull) to heavy tail (Exponential). In each case the sample size used in making the estimate was 1000 and there were 5000 repetitions.

TABLE 2a Density Function Estimate at .90 Quantile for Weibull Distribution with shape parameter 4. The actual value is .748.

	SIDDIQUI ESTIMATOR*		LEAST SQUARES ESTIMATOR	
C	AVERAGE	MSE	AVERAGE	MSE
.05	.754	.0217	.812	.0371
.10	.760	.0111	.784	.0162
.15	.731	.0067	.772	.0098
.20	.718	.0054	.767	.0073
.25	.681	.0076	.764	.0057
.30	.615	.0129	.762	.0047
.35	.572	.0328	.760	.0039
.40	.389	.1308	.737	.0026

* Optimal C from (3.34) is .229

TABLE 2b Density Function Estimate at .90 Quantile for Normal Distribution. Actual value is .176.

	SIDDIQUI ESTIMATOR*		LEAST SQUARES ESTIMATOR	
C	AVERAGE	MSE	AVERAGE	MSE
.05	.176	.00129	.191	.00209
.10	.178	.00060	.184	.00094
.15	.171	.00038	.181	.00055
.20	.168	.00032	.180	.00042
.25	.158	.00048	.180	.00034
.30	.149	.00086	.180	.00029
.35	.131	.00213	.179	.00024
.40	.085	.00829	.171	.00017

* Optimal C from (3.34) is .188

TABLE 2c Density Function Estimate at .90 Quantile for Exponential Distribution Actual value is .10.

	SIDDIQUI ESTIMATOR*		LEAST SQUARES ESTIMATOR	
C	AVERAGE	MSE	AVERAGE	MSE
.05	.100	.00043	.109	.00073
.10	.101	.00021	.106	.00032
.15	.096	.00014	.104	.00022
.20	.093	.00015	.104	.00017
.25	.086	.00027	.104	.00015
.30	.079	.00051	.105	.00013
.35	.065	.00125	.104	.00010
.40	.036	.00421	.089	.00018

* Optimal C from (3.34) is .163

5. SIMULATION FOR STOPPING TIMES: Four procedures for determining when to stop a simulation are compared using (of all things!) a simulation. Each procedure was repeated 5000 times.
The procedures were:
(a) An initial sample of size 1000 was selected. We used the Siddiqui density estimator with C=.2 and then using equation (2.1) obtained an estimate of the sample size necessary for a 95% probability of being within a prescribed distance of the .90 quantile. The required additional sample was then drawn and using the combined sample the quantile was estimated. For each repetition we record the sample size used and whether the estimate is within the prescribed distance of the actual .90 quantile.

(b) Same as (a) except that we used the Least Squares density estimator with C=.35.

(c) Same as (b) except that after the second sample is drawn we again calculate a (least squares) estimate of the density function and using equation (2.1) again, if we need a larger sample than has already been drawn we draw the required additional observations. This is called the 3 stage least squares.

(d) Starting with an initial sample size of 1000 we calculated a 95% confidence interval for the .90 quantile. If the half width of the interval was less than the prescribed distance used with the other procedures we stopped. If not we draw an additional 200 observations and using the combined sample determine again the confidence interval and again compared the half width to the previously prescribed distance. This is repeated until the half width of the interval is less than the prescribed distance. This follows the Dallal and Wilkinson (1986) procedure.

TABLE 3a Results of Stopping Time Simulation-for Weibull Distribution

Procedure	Proportion Within .015 of .90 Quantile	Average Sample Size	Standard Deviation of Sample Size
Siddiqui (a)	.9556	3056	
Least Sq (b)	.9488	2910	377
Least Sq. (c) (3 stages)	.9542	2952	345
Conf. Int (d)	.9488	2895	687

TABLE 3b Results of Stopping Time Simulations for Normal Distribution

Procedure	Proportion Within .065 of .90 quantile	Average Sample Size	Standard Deviation of Sample Size
Siddiqui (a)	.958	3016	559
Least Sq (b)	.951	2877	386
Least Sq (c) (3 stages)	.958	2910	360
Conf Int (d)	.950	2796	682

TABLE 3c Results of Stopping Time Simulations for Exponential Distribution

Procedure	Proportion Within .120 of .90 Quantile	Average Sample Size	Standard Deviation of Sample Size
Siddiqui (a)	.961	2866	586
Least Sq (b)	.939	2280	386
Least Sq (c) (3 Stages)	.944	2370	352
Conf Int (d)	.943	2552	632
Least Sq (e)*	.970	3073	509

* To indicate how sensitive the results can be to the choice of C, we also ran this with C=.4 with considerably different results than under (b) with C=.35.

In order to show what happens when the procedure causes the simulation to stop with a smaller sample size we also include Table 4. A similar table is given in Lilliefors (1987).
This table gives the breakdown of the proportion of the quantile estimates that are within the prescribed .065 of the .90 quantile for the Normal distribution according to sample size intervals (eg <1250, between 1250 and 1500, etc)

Table 4 Breakdown by sample size for Confidence Interval Procedure for normal distribution

Sample Size	number of Samples	Number Within .065 of .90 Quantile	Proportion within .065 of Quantile
1250	30	23	.77
1251-1500	38	31	.82
1501-1750	124	107	.86
1751-2000	516	468	.91
2001-2250	484	446	.92
2251-2500	620	585	.94
2501-3000	1546	1484	.96
3001-3500	853	833	.98
3501-4000	628	614	.98
>4000	161	160	.99

5. DISCUSSION OF RESULTS:
(a) First of all it should be noted that this has been a rather limited comparison. I do have some additional results for the .95 quantile which are pretty much in accord with these.

(b) One comforting result is that all the procedures seem to work reasonably well.

(c) The only real difference between the confidence interval procedure and the new procedures that are considered is in terms of the standard deviation of the sample size. The confidence interval procedure seems to have a consistently larger standard deviation than the procedures using the density estimators. As noted in Lilliefors (1987) the proportion of intervals that cover the true value of the quantile conditional on the sample size being small may be much less than the nominal 95%. See also Table 1 for another look at this problem.

(d) As noted previously, it appears that the least squares estimator is better than the Siddiqui estimator. It is generally less sensitive to the value of C (which determines the interval width) and provides an estimate with a smaller standard deviation.

References:

Bloch, D.A. and Gastwirth,J.L. (1986), "On a Simple Estimate of the Reciprocal of the Density Function." Ann Math Statist, 39, 1083-1085.

Bofinger, Eve (1975), "Estimation of a Density Function Using Order Statistics", Australian J. Statist., 17(1), 1-7.

Dallal,G.E. and Wilkinson,L. (1986), "An Analytic Approximation to the Distribution of Lilliefors's Test Statistic for Normality", The American Statistician, Vol 40, No.4, 294-296.

David, H.A. (1970), "Order Statistics", Wiley.

Eldessouky, S.A. (1985), "Stepwise Regression Using Least Absolute Value Criterion", unpublished DSc dissertation.

Law,A.M. and Kelton,W.D. (1982),"Simulation Modeling and Analysis",McGraw Hill.

Lilliefors, H. (1987), "Old Simulation Results and When to Stop",The American Statistician,41.

Siddiqui, M. (1960), "Distribution of Quantiles in Samples from a Bivariate Population, J. Res. Nat. Bur. of Standards, B64, 145-150

Silverman, B.W. (1986), "Density Estimation for Statistics and Data Analysis", Chapman and Hall.

SIMULATING STATIONARY GAUSSIAN ARMA TIME SERIES

Terry J. Woodfield, SAS Institute Inc.
Box 8000, SAS Circle, Cary, NC 27512-8000

1. INTRODUCTION

Many instructors and researchers often find that the simulation of time series data is a necessary part of their work. The proliferation of textbooks that describe the Box-Jenkins strategy for modeling time series has popularized the use of time series models having a stationary autoregressive moving average (ARMA) structure. Accurate and efficient simulation algorithms for Gaussian ARMA processes are required for many applications.

The literature on simulation of stochastic data is vast, but specific articles focusing on time series simulation are rare. The algorithms discussed in this paper have been extracted from a variety of secondary sources. Primary sources are scarce, perhaps because the algorithms are straightforward and easily derived and hence not suitable for publication in scholarly journals.

There are three components of a time series generator.

1. Algorithm to generate pseudo random numbers.
2. Algorithm to convert pseudo random numbers to pseudo random normal deviates.
3. Algorithm to convert pseudo random normal deviates to a time series.

Efficient and practical solutions to components 2 and 3 have existed for some time. Component 1 has been investigated extensively, but choice of an optimal pseudo random number generator is still an open question possibly having no unique solution. The current practice seems to be to declare a random number generator to be adequate unless it can be shown to have poor properties. Thus, our research is motivated by the concern that a pseudo random number generator that has passed existing tests may fail to produce reasonable time series data. Intuitively, pseudo random number generators that have good n-space uniformity should produce white noise sequences that are adequate for generating ARMA time series. This paper provides some preliminary results that support this conjecture.

2. THE MODEL

The univariate ARMA model for a stationary time series is

$$\phi(B)(Y_t - \mu) = \theta(B)\epsilon_t, \tag{1}$$

where

1. B is the backshift operator defined by $BY_t = Y_{t-1}$.
2. $\phi(B) = \phi_0 + \phi_1 B + \phi_2 B^2 + \ldots + \phi_p B^p$, $\theta(B) = \theta_0 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q$, where $\phi_0 = \theta_0 = 1$.
3. $\{\epsilon_t\}$ is a white noise series, i.e., independently and normally distributed with mean zero and variance $\sigma^2 > 0$.
4. The zeroes of $\phi(B)$ and $\theta(B)$ lie outside the unit circle, and $\phi(B)$ and $\theta(B)$ have no common zeroes.

The $\{\epsilon_t\}$ series is referred to as the error sequence or the innovation sequence. The notation and terminology employed is primarily that of Box and Jenkins (1976). However, note that the signs of the model coefficients are opposite those given by Box and Jenkins.

The methods described will generate a series with $\mu = E[Y_t] = 0$ and innovation variance $\sigma^2 = 1$. To obtain a time series W_t with specified mean μ and error standard deviation $\sigma_w > 0$, use the transformation $W_t = \sigma Y_t + \mu$. The variance of W_t will be σ^2 times the variance of Y_t. For specified variance σ_w^2, use transformation $W_t = kY_t + \mu$, where $k = \sqrt{\sigma_w^2/\sigma_y^2}$.

All methods may use the following recursion to generate $Y_p, Y_{p+1}, \ldots, Y_{n-1}$.

$$Y_t = -\sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}, \; t = p, p+1, \ldots, n-1. \tag{2}$$

We may treat the methods as differing only in how they produce starting values $Y_0, Y_1, \ldots, Y_{p-1}$. Note that for the two exact methods, this recursion is described in the context of the method employed and is somewhat different than is given in equation (2). There are algorithms that do not depend on the ARMA recursion relationship; for example, an efficient algorithm exists that uses the Kalman Filter. Also, some applications have employed the algorithms discussed to generate the entire series and not just the starting values.

Typically, all algorithms will have a branch such that if $p = 0$, then the simple moving average recursion is employed on a white noise sequence of length $n + q$. The algorithm is

1. Generate $\epsilon_{-q}, \epsilon_{-q+1}, \ldots, \epsilon_{n-1}$ using an appropriate random normal generator such as the one proposed by Marsaglia as described in Kennedy and Gentle (1980).
2. Form

$$Y_t = \sum_{k=0}^{q} \theta_k \epsilon_{t-k}, \; t = 0, 1, \ldots, n-1.$$

3. SIMULATION ALGORITHMS

In this work, four methods are discussed for simulating an ARMA(p,q) process.

3.1 Approximate Methods

The simplest approximation method uses starting values $Y_{-p} = Y_{-p+1} = \ldots = Y_{-1} = 0$ and employs the recursions of equation (2) to generate the series realization. Since the first few values in the simulated series will be effected by the null starting values, the point at which the effect of these values is minimal will be used as the starting point for the series.

The effect of starting values Y_{-k}, $k = 1, 2, \ldots, p$, can be monitored using the following algorithm. Let $\Phi_{t,k}$ be the coefficient representing the effect of Y_{-k} on future value Y_t, and let $\Theta_{t,j}$ be the coefficient representing the effect of ϵ_j on Y_t. The relationship between Y_t, the starting values, and the innovations is given by

$$Y_t = \sum_{k=1}^{p} \Phi_{t,k} Y_{-k} + \sum_{j=-q}^{t} \Theta_{t,j} \epsilon_j \tag{3}$$

The weights $\Phi_{t,k}$ and $\Theta_{t,k}$ are obtained using the following recursions.

$$\Phi_{0,k} = -\phi_k, \qquad k = 1, 2, \ldots, p$$

$$\Theta_{0,-k} = \theta_k, \qquad k = 1, 2, \ldots, q.$$

$$\Theta_{0,k} = 0, \qquad k = 0, 1, 2, \ldots$$

$$\Phi_{t,k} = -\phi_{t+k} - \sum_{i=1}^{t-1} \phi_i \Phi_{t-i,k}$$

$$\Theta_{t,k} = \theta_{t-k} - \sum_{i=1}^{t-1} \phi_i \Theta_{t-i,k}, \; k = -q, \ldots, 1.$$

$$\Theta_{t,k} = \Theta_{t-1,k-1}, \; k = 0, 1, 2, \ldots$$

where any coefficients with subscripts out of bounds are taken to be zero. For example, consider the model

$$(1 - 0.4B - 0.32B^2)Y_t = (1 - 0.3B - 0.1B^2)\epsilon_t.$$

The Φ matrix is given by

$$\Phi = \begin{pmatrix} 0.40000 & 0.320000 \\ 0.48000 & 0.128000 \\ 0.32000 & 0.153600 \\ 0.28160 & 0.102400 \\ 0.21504 & 0.090112 \end{pmatrix}$$

where rows are numbered $t = 0, 1, 2, 3, 4$, and columns are numbered $k = 1, 2$. The Θ matrix is given by

$$\Theta = \begin{pmatrix} -0.10000 & -0.30000 & 1.0000 & 0.000 & 0.00 & 0.0 & 0.0 \\ -0.04000 & -0.22000 & 0.1000 & 1.000 & 0.00 & 0.0 & 0.0 \\ -0.04800 & -0.18400 & 0.2600 & 0.100 & 1.00 & 0.0 & 0.0 \\ -0.03200 & -0.14400 & 0.1360 & 0.260 & 0.10 & 1.0 & 0.0 \\ -0.02816 & -0.11648 & 0.1376 & 0.136 & 0.26 & 0.1 & 1.0 \end{pmatrix}$$

where rows are numbered $t = 0, 1, 2, 3, 4$, and columns are numbered $k = -2, -1, 0, 1, 2, 3, 4$. Hence, the series $\mathbf{Y} = (Y_0, Y_1, \ldots, Y_n)'$ may be formed using

$$\mathbf{Y} = \Theta\mathbf{e} + \Phi\mathbf{y},$$

where $\mathbf{e} = (\epsilon_{-q}, \epsilon_{-q+1}, \ldots, \epsilon_n)'$, and $\mathbf{y} = (Y_{-1}, Y_{-2})'$.

Note that the last row of Θ converges to the infinite MA representation of the model. When the last row of Φ is negligible, then one may assume that the effect of the starting values has vanished and that steady state has been reached. However, when starting values of zero are employed, steady state is not reached until the last row of the Θ matrix closely matches the infinite MA representation of the model. In practice, the approximation methods are not very competitive because of the computational burden of determining Φ in order to determine where the actual series that is generated is to begin.

A more convenient approach to that of computing the Φ matrix is to obtain starting values using a truncated infinite MA representation for the ARMA(p,q) model. Let $\Psi(B) = \theta(B)/\phi(B)$ where

$$\Psi(B) = 1 + \psi_1 B + \psi_2 B^2 + \psi_3 B^3 + \ldots = \sum_{i=0}^{\infty} \psi_i B^i. \qquad (3)$$

The algorithm may be implemented as follows.

1. Find k such that $|\psi_k| \geq$ TOL and $\psi_{k+1}, \psi_{k+2}, \ldots, \psi_{k+p+q}$ are all less than TOL in absolute value for some specified tolerance TOL.
2. Generate $\epsilon_{-k}, \epsilon_{-k+1}, \ldots, \epsilon_{-1}, \epsilon_0, \epsilon_1, \ldots, \epsilon_{n-1}$ using an appropriate random normal generator.
3. Form $Y_t = \epsilon_t + \psi_1\epsilon_{t-1} + \psi_2\epsilon_{t-2} + \ldots + \psi_k\epsilon_{t-k}$, for $t = 0, 1, \ldots, p-1$.
4. Generate $Y_p, Y_{p+1}, \ldots, Y_{n-1}$ using recursion equation (2) above.

This algorithm for generating a time series will be referred to in this paper as the Psi Weight Method. Note that using starting values of zero is inferior to using starting values generated by the Psi Weight Method.

The Psi Weight Method is a useful "quick and dirty" algorithm. It may be programmed quickly in a matrix language or a lower level computer language. It requires no laborious calculations and can be speeded up using a fast finite Fourier transform algorithm.

Another approach uses a linear transformation of p white noise values based on the autocovariance function of an ARMA(p,q) process. The method is implemented as follows.

1. Obtain $\Gamma = (Cov(Y_i, Y_j)) = (\gamma_{ij})$, $0 \leq i, j \leq p-1$. For a method to compute the autocovariance function of an ARMA(p,q) process, see McLeod (1975,1977).
2. Form the Cholesky root of Γ, that is, find $\mathbf{H}$ such that $\Gamma = \mathbf{H}\mathbf{H}'$, where $\mathbf{H}$ is a lower triangular matrix.
3. Generate $\xi_0, \xi_1, \ldots, \xi_{p-1}$, using an appropriate random normal generator.
4. Form $\mathbf{Y} = (Y_0, Y_1, \ldots, Y_{p-1})'$ using $\mathbf{Y} = \mathbf{H}\mathbf{e}$, where $\mathbf{e} = (\xi_0, \xi_1, \ldots, \xi_{p-1})'$.
5. Generate $\epsilon_{p-q}, \epsilon_{p-q+1}, \ldots, \epsilon_{n-1}$, using an appropriate random normal generator.
6. Generate $Y_p, Y_{p+1}, \ldots, Y_{n-1}$ using recursion equation (2) above.

This algorithm will be called the Approximate Autocovariance Method in this work.

Note that an equivalent but computationally more intensive method can be based on steps 1 through 4 using an n by n covariance matrix rather than a p by p covariance matrix. The method using the full n by n covariance matrix is exact. Otherwise, this method is not exact because the error sequence $\mathbf{e} = (\xi_0, \xi_1, \ldots, \xi_{p-1})'$ is independent of the innovation sequence $\epsilon_{p-q}, \epsilon_{p-q+1}, \ldots, \epsilon_{n-1}$. The innovations used to compute $Y_{p+1}, Y_{p+2}, \ldots, Y_{p+q}$ do not take into account the covariance $E(Y_{t+k}\epsilon_t)$ between the innovations and the time series. To get an exact ARMA(p,q) realization, the first q innovations must be generated having the appropriate covariance structure with the time series. A method that accomplishes this task is described below.

3.2 Exact Methods

An exact finite realization of an ARMA(p,q) process can be obtained using an enhanced version of the linear transformation algorithm employed above. The method is implemented as follows.

1. Obtain $\Gamma_0 = (Cov(Y_i, Y_j)) = (\gamma_{ij})$, $0 \leq i, j \leq p-1$.
2. Let $m = \max(1, q - p + 1)$, and obtain psi weights $\psi_1, \psi_2, \ldots, \psi_{q-m}$ given by equation (3) above. Form the matrix

$$\Psi = \begin{pmatrix} \psi_0 & 0 & 0 & \ldots & 0 \\ \psi_1 & \psi_0 & 0 & \ldots & 0 \\ \psi_2 & \psi_1 & \psi_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \psi_{q-m} & \psi_{q-m-1} & \psi_{q-m-2} & \ldots & \eta \end{pmatrix}$$

where $\psi_0 = 1$ and $\eta = 1$ if $m = 1$, $\eta = 0$ otherwise. If $p > q$ place $p - q$ rows of zeroes at the beginning of Ψ so that

$$\Psi = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \psi_0 & 0 & 0 & \ldots & 0 \\ \psi_1 & \psi_0 & 0 & \ldots & 0 \\ \psi_2 & \psi_1 & \psi_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \psi_{q-m} & \psi_{q-m-1} & \psi_{q-m-2} & \ldots & \eta \end{pmatrix}$$

3. Form the covariance matrix

$$\Gamma_1 = \begin{pmatrix} \Gamma_0 & \Psi \\ \Psi' & \mathbf{I}_q \end{pmatrix}$$

where $\mathbf{I}_q$ is the q by q identity matrix.
4. Form the Cholesky root $\mathbf{H}$ of Γ_1.
5. Generate $\epsilon_{-q}, \epsilon_{-q+1}, \ldots, \epsilon_{p-1}$, using an appropriate random normal generator.

6. Form $\mathbf{S} = (S_{-q}, S_{-q+1}, \ldots, S_{p-1})'$ using $\mathbf{S} = \mathbf{He}$, where $\mathbf{e} = (\epsilon_{-q}, \epsilon_{-q+1}, \ldots, \epsilon_{p-1})'$. Let

$$\mathbf{W} = (S_{-q+p}, S_{-q+p+1}, \ldots, S_{p-1}, \epsilon_p, \ldots, \epsilon_{n-1})'.$$

7. Form

$$X_t = \sum_{k=0}^{q} \theta_k W_{p+t-k}, \ t = 0, 1, \ldots, n-p-1.$$

8. Form

$$Y_t = S_{t-q}, \ t = 0, 1, \ldots, p-1,$$
$$Y_t = X_{t-p} - \sum_{k=1}^{p} \theta_k Y_{t-k}, \ t = p, p+1, \ldots, n-1.$$

Thus, the values $Y_0, Y_1, \ldots, Y_{p-1}$ come directly from the transformation using the Cholesky factorization, $Y_p, Y_{p+1}, \ldots, Y_{p+q-1}$ come from the recursion relation and the Cholesky factorization of the expanded covariance matrix, and $Y_{p+q}, Y_{p+q+1}, \ldots, Y_{n-1}$ come from the recursion relation. In this paper, the above algorithm is called the Exact Autocovariance Method. The Exact Autocovariance Method is probably the most popular simulation technique used in practice. For example, Ansley and Newbold (1980) describe use of the algorithm in an appendix, and an example of the algorithm programmed in SAS/IML® by David M. DeLong is given in the SAS/IML User's Guide (1985a).

The last method to be considered is described in a homework exercise in Brockwell and Davis (1987, page 264, problem 8.17) and is based on the *Innovations Algorithm* used in finite memory prediction. The method is related to the linear prediction approach suggested by Wilson (1978). The method is implemented as follows.

1. Generate $\epsilon_0, \epsilon_1, \ldots, \epsilon_{n-1}$ using an appropriate random normal generator.
2. Let $m = \max(p, q)$. Form

$$W_t = \sigma_t \epsilon_t + \sum_{i=1}^{t} \theta_{t,i} \sigma_{t-i} \epsilon_{t-i}, \ t \le m-1,$$
$$= \sigma_t \epsilon_t + \sum_{i=1}^{q} \theta_{t,i} \sigma_{t-i} \epsilon_{t-i}, \ t \ge m,$$

where the coefficients θ_{ti} and σ_t are obtained using the recursion

$$\sigma_0^2 = \Gamma(1,1),$$
$$\theta_{n,n-k} = \left(\Gamma(n+1, k+1) - \sum_{i=0}^{k-1} \theta_{k,k-i} \theta_{n,n-i} \sigma_i^2\right) / \sigma_k^2,$$
$$k = 0, 1, \ldots, n-1,$$
$$\sigma_n^2 = \Gamma(n+1, n+1) - \sum_{i=0}^{n-1} \theta_{n,n-i}^2 \sigma_i^2,$$

with the covariance function $\Gamma(i,j)$ defined by

$$\begin{aligned} \Gamma(i,j) &= \gamma_{i-j}/\sigma^2, && 1 \le i,j \le m, \\ &= \left(\gamma_{i-j} + \textstyle\sum_{k=1}^{p} \phi_k \gamma_{k-|i-j|}\right)/\sigma^2, && \\ &\quad \min(i,j) \le m < \max(i,j) \le 2m, && \\ &= \textstyle\sum_{k=0}^{q} \theta_k \theta_{k+|i-j|}, && \min(i,j) > m, \\ &= 0 && \text{otherwise} \end{aligned}$$

The values $\{\gamma_k\}$ used to compute $\Gamma(i,j)$ are the autocovariances for the original ARMA(p,q) process, and σ^2 is the error variance.

3. Form $Y_0, Y_1, \ldots, Y_{n-1}$ using

$$Y_t = \sigma W_t, \ t \le m,$$
$$= \sigma W_t + \sum_{k=1}^{p} \phi_k Y_{t-k}, \ t > m.$$

4. RANDOM NUMBER GENERATION

The algorithms discussed above employ an independent normal random deviate generator. The algorithm of Marsaglia as described in Kennedy and Gentle (1980) has been mentioned as a suitable normal deviate generator. Since Marsaglia's algorithm is efficient and provides independent deviates that have an exact normal distribution, we will not consider competing exact or approximate methods. Instead, we will focus on pseudo random number generators that may be potential candidates for use with Marsaglia's normal deviate generator.

A linear congruential pseudo random number generator employs the recursion

$$X_n = mX_{n-1} + c (\text{mod } M)$$

When $c = 0$, the generator is called a multiplicative congruential pseudo random number generator. The constant m is called the multiplier, and M is called the modulus. The choice of multiplier m determines the properties of the generator. Fishman and Moore (1982, 1986) evaluate the generator for the most commonly suggested multipliers.

The Tausworthe pseudo random number generator is a special case of the Generalized Feedback Shift Register (GFSR) algorithm (Lewis and Payne 1973). Kennedy and Gentle (1980) provide computer code for a simple Tausworthe generator that uses the primitive polynomial $p(x) = x^{31} + x^5 + x^2 + 1$.

For details on pseudo random number generators and how they may be tested, see Kennedy and Gentle (1980) or Knuth (1981). Note that for any pseudo random number generator, a starting value X_0 is required. For some generators, choice of starting value has an effect on the properties of the pseudo random sequence produced. A number of theoretical and empirical tests exist to evaluate random number generators. We will evaluate several generators with respect to the quality of ARMA time series produced using the generators.

5. EVALUATING THE ALGORITHMS

For algorithms that depend on recursion (2), the sequences produced will converge to a common time series if a common error sequence is used in the recursion. Using early values before the time series has reached steady state is not recommended. Assuming that any implementation of an ARMA time series generator will ensure that only steady state values are used, any other comparison of the algorithms should be based only on numerical properties related to their implementation on finite precision digital computers.

The behavior of any given time series generator ultimately depends on the choice of uniform random number generator to be employed. Hence, statistical evaluation of time series generators will be carried out using a designed experiment involving type of uniform generator as a primary factor of interest. Given the equivalence of the algorithms after steady state is reached, the experiment described in the next section will employ only the Exact Autocovariance Method.

To measure the quality of a generated time series, goodness-of-fit criterion are developed. One approach to measuring goodness-of-fit makes comparisons between the sample autocovariances and the true autocovariances. Two measures of closeness are

$$MSE = \frac{1}{m}\sum_{k=0}^{m-1}(\gamma_k - \hat{\gamma}_k)^2,$$

$$MAD = \frac{1}{m}\sum_{k=0}^{m-1}|\gamma_k - \hat{\gamma}_k|,$$

where γ_k is the theoretical autocovariance function at lag k, $\hat{\gamma}_k$ is the sample autocovariance function at lag k, and m is chosen so that values of the true autocovariance sequence beyond lag m are relatively small. Intuitively, the sample autocovariances will converge to the true autocovariance function, so the measures MSE and MAD will reflect whether the series generated comes from the specified model. As series length increases, MSE and MAD should get smaller. Hence, the use of MSE and MAD is more meaningful for evaluating the quality of long generated series. If MSE and MAD do not approximate a monotone decreasing sequence for increasing n, then the simulation algorithm employed is unacceptable.

One problem with the autocovariance approach to measuring goodness-of-fit is that the values MSE and MAD may not provide a true measure of closeness for small series because the autocovariance estimator is biased. Thus, it is conceivable that a generated series may be better than another series in some sense, but may produce a biased estimate of the autocovariance function that is worse than the biased estimate produced by the other sequence. While it is unlikely that the bias will uniformly favor one method over another even when that method is inferior by most criterion one might devise, one should nonetheless be made aware of the potential pitfalls in this method of comparison. If significant differences are noted, one explanation is that the better method produces sequences that minimize bias in estimation.

Note also that for fixed series length n, MSE or MAD may favor a series that is generated to have smaller innovation variance than another series. This is primarily because MSE or MAD may be small when $\hat{\gamma}_k$ is close to zero. On the other hand, MSE and MAD values will tend to remain constant in such cases across all sample sizes, whereas series generated using the correct innovation variance will experience MSE and MAD values that get smaller with increased series length. This fact also negates the potential negative effect of bias in measuring goodness-of-fit, since a faulty generator is unlikely to produce time series with biased autocovariance estimates that uniformly beat the estimates produced by a better generator across all sample sizes. Recall that for large sample sizes, the bias factor tends to be relatively small.

Other approaches exist for measuring goodness-of-fit of generated time series. Most approaches will have the same pitfalls as those discussed above. The numerical overhead required in implementing many goodness-of-fit measures places a severe penalty on their use in large scale simulation studies.

6. A MONTE CARLO EXPERIMENT

An experiment has been performed to investigate properties of time series generators. The factor of interest in the experiment is the type of uniform generator employed. There are six levels for factor 1 corresponding to six algorithms for generating pseudo random numbers. The six pseudo random number generators considered are

1. multiplicative congruential generator with $m = 65539$
2. multiplicative congruential generator with $m = 16807$
3. multiplicative congruential generator with $m = 397204094$.
4. multiplicative congruential generator with $m = 742938285$.
5. multiplicative congruential generator with $m = 32$.
6. Tausworthe generator (Kennedy and Gentle, 1980, p. 155).

The modulus for generators (1) through (5) is $M = 2^{31} - 1$. Hence, these generators are appropriate for computers having 32 bit words. Note that generators (1) through (4) have been investigated by Fishman and Moore (1982, 1986). Generator (1) is the notorious RANDU generator. Generators (2) and (3) are available in IMSL® (1987). Generator (3) is also available in SAS (1985b). Fishman and Moore (1986) suggest that generator (4) is superior to generators (1) through (3). Generator (5) is known to have poor runs properties and is included as a control. If generator (5) cannot be judged significantly worse than the other generators, then one should look for flaws in the Monte Carlo experiment and at the very least view the results with caution.

In addition, two factors sample size and ARMA model employed, are required to attempt to generalize the results to a wide variety of situations likely to be encountered in practice. The sample sizes considered are $n = 50$, 100, and 500. The models employed are:

1. Ansley and Newbold (1981)

$$(1 - 0.80B + 0.65B^2)Y_t = \epsilon_t.$$

2. Ansley and Newbold (1981)

$$Y_t = (1 + 1.25B + 0.35B^2)\epsilon_t$$

3. Ansley and Newbold (1981)

$$(1 - 0.95B)Y_t = (1 + 0.85B)\epsilon_t$$

4. Woodward and Gray (1981)

$$(1 - 1.5B + 1.21B^2 - 0.455B^3)Y_t = (1 + 0.2B + 0.9B^2)\epsilon_t$$

5. Brockwell and Davis (1987)

$$(1 - B + 0.24B^2)Y_t = (1 + 0.4B + 0.2B^2 + 0.1B^3)\epsilon_t$$

6. *Newton and Pagano (1983)*

$$(1 - 0.3357B + 0.0821B^2 + 0.1570B^3 + 0.2567B^4)Y_t = (1 - 0.6077B + 0.0831B^2 + 0.1903B^3)\epsilon_t$$

Ten replications were performed for each factor level combination. In all cases, seeds were transmitted from one routine to the next with no attempt to control seed values or synchronize the series generated. The response variables are MSE and MAD defined above for the autocovariance function. The value m used to truncate the autocovariance sequence was chosen so that $|\gamma_k| < 0.00001$ for all $k > m$, except for the pure MA model 2, in which case m was arbitrarily set to ten. Table 1 lists the values of m (truncation lag value) and values of MSE (mean sum of squares) and MAD (mean absolute value) with $\hat{\gamma}_k$ set equal to zero for all k.

Tables 2 and 3 provide listings of the generators that produced the lowest or highest cell mean for the given sample size by model combination. While the results for the lowest cell mean do not appear to be randomly dispersed in the table, the table summarizing the number of times each generator had lowest cell mean provides values that will not lead to rejection of a null hypothesis that the lowest cell mean is distributed uniformly across generators.

More insight is obtained from table 3. The results clearly reject uniformity and strongly imply that generator 5 is inadequate. Generators 4 and 6 also have an unusually high number of counts, although the experiment is too small to allow one to draw any strong conclusions.

Table 4 also provides strong evidence that generator 5 is inadequate given that it fails in several situations to adequately generate series that exhibit the monotone decreasing behavior of the response variables. The non-starred items in table 4 represent cases where monotonicity was violated, but only for sample sizes of 50 and 100. The non-starred items also reflect small increases that are probably a result of sampling error rather than generator deficiencies.

Since the Monte Carlo results may not satisfy the assumptions to carry out the usual parametric ANOVA, a nonparametric ANOVA was performed based on replacing response values by their Blom normal scores. Initially, all F tests exhibited p-values smaller than 0.01. Examination of the cell means revealed that model 3, having roots near the unit circle, produced MSE or MAD values considerably higher than those for other models. As indicated above, generator 5 also consistently produced unusually high values for most models. When model 3 and generator 5 were deleted from the study, the model by generator and model by sample size interactions were significant at the five percent level. However, there is not enough evidence to reject any of the remaining generators as being inadequate. If all generators are basically of equal quality, it is not surprising that a statistically significant model by generator interaction is observed.

Finally, since the response variables MSE and MAD are more appropriate for larger sample sizes, we carried out the ANOVA for the case $n = 500$. Both response variables lead to the conclusion that there is a statistically significant interaction between model and generator. In the presence of interaction we can only draw conclusions about the effect of the generators for the particular models in the study. There is no compelling evidence to imply that any of the five remaining generators may be consistently superior or inferior to the others.

Most results were consistent whether MSE or MAD was used as a criterion measure. Any discrepancies may have been due to MSE being more sensitive to outliers than MAD. Results also supported the consistency of the sample autocovariances. In this regard, only generator (5) can be declared unacceptable by this analysis. While the RANDU generator is generally considered to be inadequate, it was not rejected in our study. This is not surprising because RANDU has been found to be adequate for many specific applications that are not adversely effected by RANDU's poor n-space uniformity properties. Further research is warranted using a variety of criterion measures and a larger experimental design.

7. CONCLUDING REMARKS

When choosing an algorithm for simulating a stationary Gaussian ARMA time series, theoretical considerations narrow the choice to efficient exact algorithms, although the Psi Weight Method is ideal for providing a quick method for generating time series in almost any computing environment. Choice of pseudo random number generator becomes the crucial problem in designing a simulation routine. A Monte Carlo study provides evidence to indicate that some of the more popular multiplicative congruential generators may be adequate for simulating time series data. Killam (1987) indicates that the non-statistical tests used by Fishman and Moore (1986) may not be as meaningful for generators employed in many statistical applications. Our preliminary work supports this view.

For future study, known properties of statistical estimators should be investigated with data simulated using the algorithms discussed in this paper. A larger study employing more generators is warranted. Only when expected behavior is observed to within an acceptable tolerance should the algorithms then be used to gain insight into statistical procedures that do not have adequate theoretical underpinnings.

Simulations described in this paper were carried out on an Apollo workstation using the SAS System and SAS/IML software, Version 6.03.

REFERENCES

Ansley, Craig F., and Newbold, Paul (1980). Finite sample properties of estimators for autoregressive moving average models. *Journal of Econometrics*, **13**, 159-183.

Ansley, Craig F., and Newbold, Paul (1981). On the Bias in Estimates of Forecast Mean Square Error. *Journal of the American Statistical Association*, **76**, 569-578.

Box, G.E.P., and Jenkins, G.M. (1976). *Time Series Analysis: Forecasting and Control*. Oakland, California: Holden-Day.

Brockwell, Peter J., and Davis, Richard A. (1987). *Time Series: Theory and Methods*. New York: Springer-Verlag.

Fishman, George S., and Moore, Louis R. (1982). A Statistical Evaluation of Multiplicative Congruential Random Number Generators with Modulus $2^{31} - 1$. *Journal of the American Statistical Association*, **77**, 129-136.

Fishman, George S., and Moore, Louis R. (1986). An Exhaustive Analysis of Multiplicative Congruential Random Number Generators with Modulus $2^{31} - 1$. *SIAM Journal of Scientific and Statistical Computation*, **7**, 24-45.

IMSL® (1987). *STAT/LIBRARY™ User's Manual*. Houston: IMSL.

Kennedy, William J., Jr., and Gentle, James E. (1980). *Statistical Computing*. New York: Marcel Dekker.

Killam, Bart (1987). An Overview of the SAS® System Random Number Generators. Proceedings of the Twelfth Annual Conference, SAS Users Group International, Dallas, Texas, 1059-1065.

Knuth, Donald E. (1981). *The Art of Computer Programming, 2nd Edition. Volume 2: Seminumerical Algorithms*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Lewis, T.G., and Payne, W.H. (1975). Generalized Feedback Shift Register Pseudorandom Number Algorithm. *Journal of the Association for Computing Machinery*, **20**, 456-468.

McLeod, Ian (1975). Derivation of the Theoretical Autocovariance Function of Autoregressive-Moving Average Time Series. *Applied Statistics*, **24**, 255-256.

_____ (1977). Correction to McLeod (1975). *Applied Statistics*, **26**, 194.

Newton, H. Joseph, and Pagano, Marcello (1983). The Finite Memory Prediction of Covariance Stationary Time Series. *SIAM Journal of Scientific and Statistical Computing*, **4**, 330-339.

Priestley, M.B. (1981). *Spectral Analysis and Time Series*. New York: Academic Press.

SAS Institute Inc. (1985a). *SAS/IML® User's Guide, Version 5 Edition*. Cary, North Carolina: SAS Institute Inc., 73-75.

SAS Institute Inc. (1985b). *SAS® Language Guide for Personal Computers, Version 6 Edition*. Cary, North Carolina: SAS Institute Inc.

Wilson, G.T. (1978). Some Efficient Computational Procedures for High Order ARMA Models. *Journal of Statistical Computation and Simulation*, **8**, 301-309.

Woodward, Wayne A., and Gray, H.L. (1981). On the Relationship Between the S Array and the Box-Jenkins Method of ARMA Model Identification. *Journal of the American Statistical Association*, **76**, 579-587.

1. True Autocovariance Function Summary Values

Model	Truncation Lag Value	Mean Sum of Squares	Mean Absolute Value
1	55	0.16742	0.15111
2	10	1.01794	0.47225
3	294	42.83101	2.38645
4	56	6.14362	0.73537
5	29	4.86766	0.98871
6	49	0.02877	0.04132

2. Summary Tables From Monte Carlo Experiment

Table entry is the number of the generator with lowest mean MSE.

n\ model	1	2	3	4	5	6
50	5	1	4	6	6	5
100	5	1	2	2	6	3
500	3	4	2	1	3	2

Table entry is the number of the generator with lowest mean MAD.

n\ model	1	2	3	4	5	6
50	5	1	4	6	6	5
100	5	1	2	2	6	3
500	3	4	2	1	2	3

Table entry is the number of times the given generator had lowest mean MSE or MAD.

generator:	1	2	3	4	5	6
	3	4	3	3	2	3

3. Summary Tables From Monte Carlo Experiment

Table entry is the number of the generator with highest mean MSE. (When 5 is the generator with highest MSE, the number of the generator with next highest MSE is given in parenthesis.)

n\ model	1	2	3	4	5	6
50	2	5(3)	5(6)	5(4)	5(2)	4
100	4	5(6)	5(6)	5(4)	5(1)	6
500	5(4)	5(2)	5(3)	5(4)	5(1)	5(1)

Table entry is the number of the generator with highest mean MAD. (When 5 is the generator with highest MAD, the number of the generator with next highest MAD is given in parenthesis.)

n\ model	1	2	3	4	5	6
50	2	5(3)	5(6)	5(4)	5(2)	4
100	6	5(6)	5(6)	5(4)	5(1)	6
500	5(4)	5(2)	5(3)	5(4)	5(1)	5(1)

Table entry is the number of times the given generator had highest mean MSE or MAD. (Value in parenthesis is number of times if generator 5 is omitted.)

generator:	1	2	3	4	5	6
MSE:	0(3)	1(3)	0(2)	2(6)	14	1(4)
MAD:	0(3)	1(3)	0(2)	1(5)	14	2(5)

4. Cases Where MSE or MAD Were not Monotone Decreasing

model	generator	model	generator	model	generator
1	3	3	5 *	5	5 *
1	4	3	6 (MSE)	6	2 (MAD)
1	6	4	5 *	6	5 (MAD)
2	6	4	6	6	6

* In all cases except for the starred items, the case n=500 had smallest MSE and MAD values.

ON COMPARATIVE ACCURACY OF MULTIVARIATE NONNORMAL RANDOM NUMBER GENERATORS

Lynne K. Edwards, University of Minnesota

Abstract

There are two easily accessible methods of generating multivariate nonnormal distributions using the IMSL. They are: a multivariate extension of a power method with an intermediate correlation matrix adjustment and a normal-mixture method. Neither these methods can produce all possible combinations of marginal skew and kurtosis, but they have an advantage over the known extreme distributions when a multivariate nonnormal distribution with specified intercorrelations and specified marginal moments is desired for simulating a plausible nonnormal situation. The MSE and se(MSE) for the four marginal moments and for the intercorrelations were compared between the two methods. The Fleishman-type method produced a much smaller bias in correlation coefficients than the normal-mixture method but the reversed trends were found for the marginal skewness and kurtosis.

Keywords: simulation algorithms; empirical moments; MSE

1. Introduction

Multivariate nonnormal random numbers are sometimes generated to simulate a realistic nonnormal distribution with specified four marginal moments and intercorrelations. As in the case of univariate nonnormal distributions, the known multivariate nonnormal distributions have highly desirable properties, such as density functions. But they are often far from the plausible nonnormal distributions that are encountered in testing and experiments. A statistic may be robust for all practical purposes under plausible nonnormality conditions, while it may exhibit nonrobustness under extremely nonnormal conditions which are almost never encountered in real studies.

The extension of known extreme univariate distributions to multivariate distributions is an obvious option but it has several shortcomings. An extreme distribution provides implausible skew and kurtosis and it is often difficult to specify the desired intercorrelations among the variates. For example, a log-normal distribution with $\mu = 0$ and $\sigma^2 = 1$ is often used as a right-skewed nonnormal distribution but it has skewness of 6.18 and kurtosis of 113.94, a far departure from a typical skewed distribution found in psychological and educational research with skewness of 0.7-1.0 and kurtosis of 2.0-5.0. Another nonnormal distribution frequently used in simulations is a Laplace, but a "cusp" is almost never found in real data. Still another frequently used nonnormal distribution is a chi-square distribution with degrees of freedom ranging from 2 to 3. Although it can be rescaled, limited combinations of skewness and kurtosis can be generated.

Various algorithms such as the Johnson-system, Burr-system, and Shmeiser-Deutch system for generating flexible distributions are well established for the univariate disributions (Rubinstein, 1981; Burr, 1973; Tadikamalla, 1980; Schmeiser & Deutch, 1977). Although the multivariate extensions of such distributions are discussed in Johnson (1987), they tend to be computationally involved.

An alternative approach is to extend the univariate normal-mixture method. This method has an intuitive appeal because we can think of a marginally skewed dataset either in test scores or in a repeated measures design, as the data obtained from three distinct subpopulations; each with a normal distribution with a different mean and a variance but with the same correlation matrix. One of the drawbacks of this method is that it may generate multi-modal distributions.

Yet another approach is to use an approximate distribution, an extension of Fleishman's univariate power method (Fleishman, 1978), to a multivariate situation. Vale and Maurelli (1983) have shown that the multivariate extension works reasonably well with their intermediate correlation adjustment.

The last two methods provide an intuitively simple extension from the univariate to the multivariate in simulating testing situations where three parallel tests are given to the same subjects, or in general when the same subjects are repeatedly observed.

The purpose of this study is to compare the Fleishman-type power method and a normal-mixture method, two relatively easy methods of simulating multivariate nonnormal distributions with specified moments and intercorrelations. It is of interest to test their relative accuracy on the marginal moments and intercorrelations.

2. A Power Method

A multivariate extension of the Fleishman's power method (Vale & Maurelli, 1983) is as follows:

Univariate procedure: $Y = a + bx + cx^2 + dx^3$

1. Solve the following nonlinear equations (Fleishman, 1978).

$$b^2 + 6bd + 2c^2 + 15d^2 - \sigma^2 = 0$$

$$2c(b^2 + 24bd + 105d^2 + 2) - \gamma_1 = 0$$

$$24[bd + c^2(1 + b^2 + 28bd) + d^2(12 + 48bd + 141c^2 + 225d^2)] - \gamma_2 = 0$$

$$a = -c \text{ if } \mu = 0$$

2. Generate a unit normal variate, **x**, for each variate needed.

3. Solve for the intermediate correlation matrix in **x**'s for the obtained coefficients: a, b, c, and d. The matrix elements below and the polynomial function for solving for the correlations in **x**'s to be specified are fully reported in Vale and Maurelli (1983), but the conditional expectations can be easily applied to solve for them. The intermediate correlations are typically larger than the specified values to counteract the attenuation in correlations resulting from the power transformation of **x**'s.

$$R = E(x_1, x_2') = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & \rho_{X1X2} & 0 & 3\rho_{X1X2} \\ 1 & 0 & \rho^2_{X1X2}+1 & 0 \\ 0 & 3\rho_{X1X2} & 0 & 6\rho^3_{X1X2}+9\rho_{X1X2} \end{bmatrix}$$

$$r_{Y1Y2} = E(Y_1, Y_2') = E(w'_1 x_1 x'_2 w_2) = w'_1 R w_2$$
$$= \rho_{X1X2}(b_1 b_2 + 3b_1 d_2 + 3d_1 b_2 + 9d_1 d_2) + \rho^2_{X1X2}(2c_1 c_2) + \rho^3_{X1X2}(6d_1 d_2)$$

4. Apply a triangular decomposition (Cholesky's) to the obtained intermediate correlation matrix R, and produce $\mathbf{x}^* = \mathbf{x}\, L'$ where $LL' = R$.

5. Apply the coefficients a,b, c, and d to $\mathbf{x}^*$.

$Y = \mathbf{w}'\mathbf{x}^*$, where $\mathbf{w}' = [a,b,c,d]$ and

$\mathbf{x}^{*'} = [1, x, x^2, x^3]$

3. A Normal-Mixture Method

A multivariate extension of a normal-mixture method (Everitt & Hand, 1981) is as follows:

1. Generate a multivariate normal distribution: Normal $p(x; \mu, \Sigma)$ $\mathbf{x} \sim N(\mu, \Sigma)$

2. For a symmetric leptokurtic distribution, solve for $\Pi_1, \Pi_2, \Sigma_1, \Sigma_2$ and for a skewed distribution, solve for $\Pi_1, \Pi_2, \Pi_3, \mu_1, \Sigma_1, \mu_2, \Sigma_2$ and μ_2, Σ_2.

Tailed $p(x; \mu, \Sigma) = \Pi_1\, p(x_1; \mu, \Sigma_1) + \Pi_2\, p(x_2; \mu, \Sigma_2)$.

Skewed $p(x; \mu, \Sigma) = \Pi_1\, p(x_1; \mu_1, \Sigma_1) + \Pi_2\, p(x_2; \mu_2, \Sigma_2) + \Pi_3\, p(x_3; \mu_3, \Sigma_3)$.

3. If a multi-modal distribution is to be avoided, a sufficient condition has to be satisfied (Everitt & Hand, 1981).

$$(\mu_2 - \mu_1)^2 < 27\,\sigma^2_1\sigma^2_2 / 4(\sigma^2_1 + \sigma^2_2)$$

Table 1. The parameters for the mixed distributions used

type	marginal μ	marginal σ^2	marginal γ_1	marginal γ_2	components Π	components μ	components σ^2
norm	0	1	0.00	0.00	1.00	0.0	1.00
tailed	0	1	0.00	3.1212	0.80	0.0	0.49
					0.20	0.0	3.04
skew	0	1	1.062	2.4366.	0.33	-0.4	0.25
					0.33	-0.2	0.25
					0.33	0.6	1.94

4. Data Generation

Two plausible nonnormal distributions with specified moments (Table 1) were generated with the IMSL (IMSL, 1986) by a power method and a normal-mixture method. The correlations were set to $\rho_{12} = 0.79$, $\rho_{13} = 0.53$, and $\rho_{23} = 0.73$. Although a univariate unit normal can be used for generating x's in the power method, the multivariate unit normal, GGNSM, was used in order to reduce variances in comparing the two methods. For a normal-mixture method, GGNSM, GGBN, and GGMTL were used to generate the mixed distributions.

The first simulation was conducted with the Cyber 855, simulating N=2000 with 50 replications. Because a negative bias in kurtosis and huge MSE's in kurtosis and skewness in the power method were noted, 50 new independent simulations were conducted with the Cray 2/4 to ascertain the accuracy of the results and to obtain the se(MSE)'s (Table 2). These 50 independent simulations represent 50 independent sets of N=2000 with 50 replications each. Randomly chosen 50 seeds were used to generate these independent simulation sets across four distributions. The figures reported in Table 2 represent the average of such independent experiments. The se(MSE) is the variability of simulations across 50 independent experiments.

5. Simulation Results

Both Fleishman's and mixture methods are limited in the type of nonnormal distributions they can generate. However, they are easy to use with the help of the IMSL and are of reasonable accuracy for researchers needing plausible nonnormal distributions with specified marginal moments up to the fourth, and with specified intercorrelations.

Within the limits of the distributions tested:
1. Fleishman's method is superior to a mixture method in generating the data with intercorrelations close to the population values and with smaller MSE's. In particular, a mixture method produced highly positively biased intercorrelations when each marginal was skewed in the same direction.

2. A normal-mixture method, a three-distribution mix for a skewed distribution, and a two-distribution mix for a tailed distribution, was superior to Fleishman's method in generating data closer to the specified skewness and kurtosis and with smaller MSE's.

It is understandable that a power method produced a set of intercorrelations close to the specified values because of the intermediate correlation adjustment. If a robustness study involves a statistic which is highly dependent on the sample intercorrelations, a power method may be more desirable than a normal-mixture method. On the other hand, if specified skewness and kurtosis accompanied by small MSE are required in a simulation, a normal-mixture method which has finite higher moments for each component normal produces smaller biases and variances and is more stable across independent simulations as indicated by small se(MSE)'s.

6. Acknowledgment

The generous computer funds from the Minnesota Supercomputer Institute and from the University of Minnesota Academic Computing Center are duly acknowledged. Special thanks are due Gyenam Kim for her assistance in programming. The author thanks Bruce Schmeiser and Tina Song for valuable suggestions and criticisms. The last but not least, the author thanks Betty Jo Johnson for editing the earlier draft.

7. References

Burr, I. W. (1973). Parameters for a general system of distributions to match a grid of $\alpha 3$ and $\alpha 4$. Communications in Statistics, 2, 1-21.

Everitt, B. S., & Hand, D. J. (1981). Finite mixture distributions. New York: Chapman & Hall.

Fleishman, A. I. (1978). A method for simulating non-normal distributions. Psychometrika, 43, 521-532.

IMSL (1986). IMSL user's guide. Houston: Author.

Johnson, M. E. (1987). Multivariate statistical simulation. New York: Wiley.

Rubinstein, R. Y. (1981). Simulation and the Monte Carlo method. New York: Wiley.

Schmeiser, B. W., & Deutch, S. J. (1977). A versatile four parameter family of probability distributions suitable for simulation. AIIE Transactions, 9, 176-182.

Tadikamalla, P. R. (1980). On simulating non-normal distributions. Psychometrika, 45, 273-279.

Vale, C. D., & Maurelli, V. A. (1983). Simulating multivariate nonnormal distributions. Psychometrika, 48, 465-471.

Figure 1. Mean

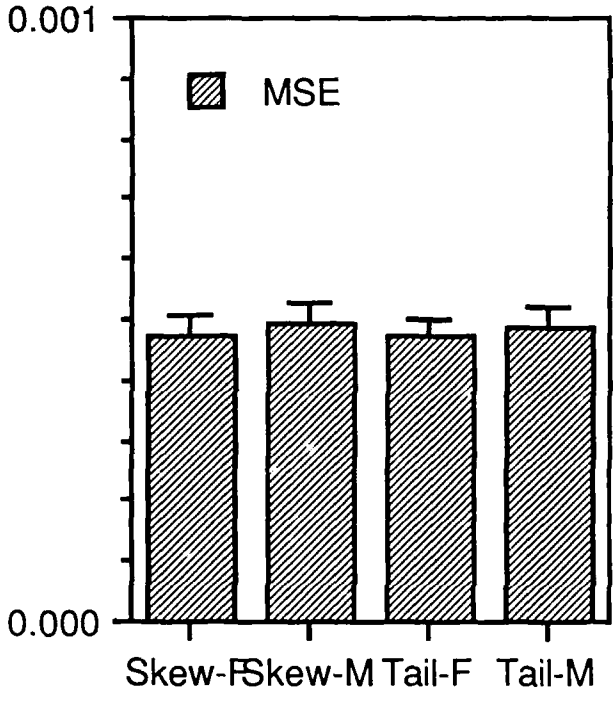


Figure 2. Variance

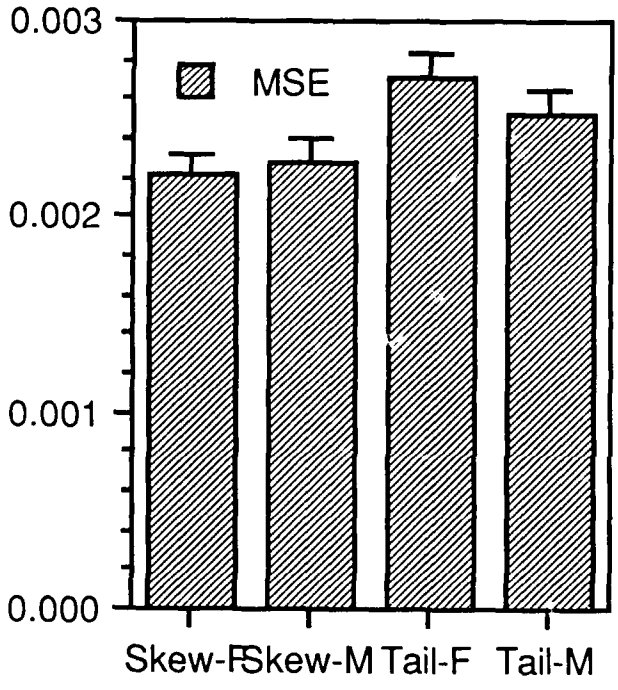


Figure 3. Skewness

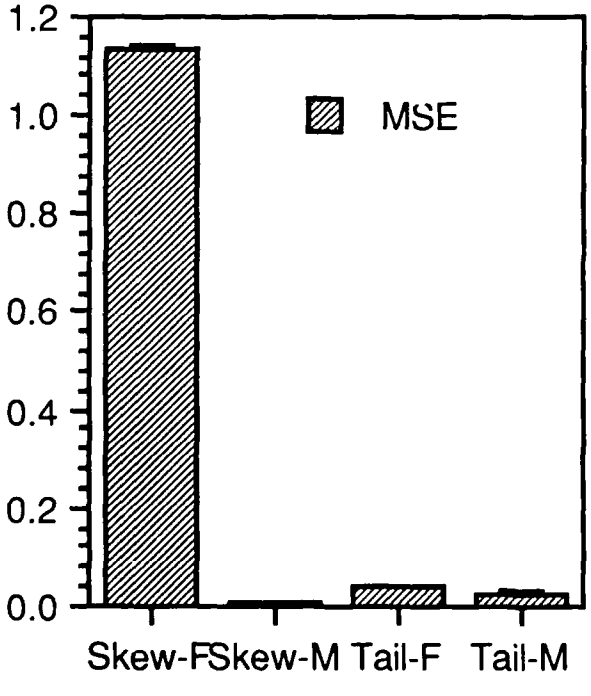


Figure 4. Kurtosis

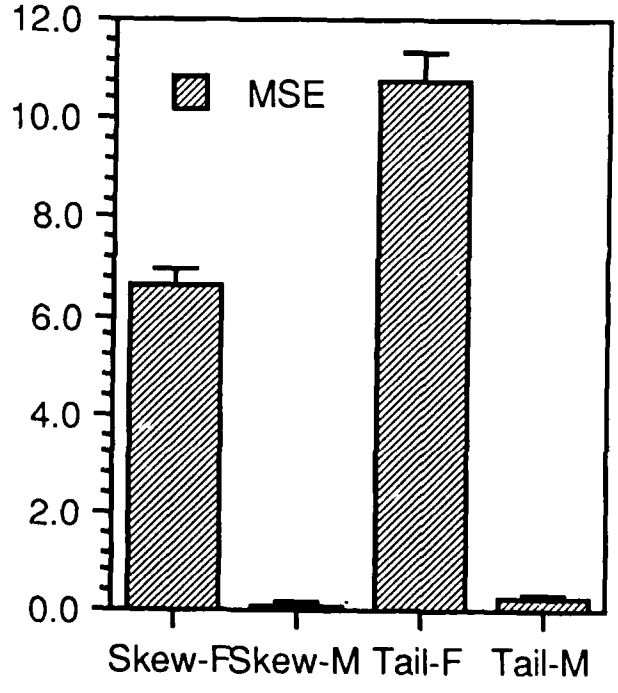


Figure 5. Correlations in a Skewed Distribution

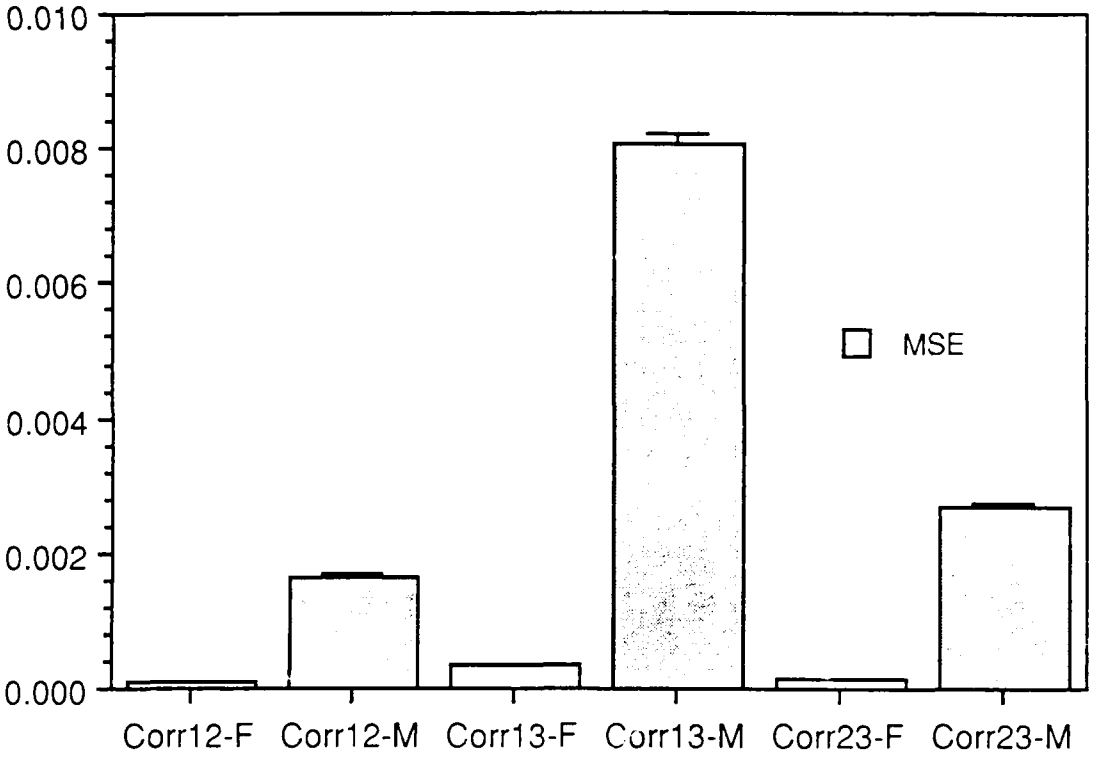


Figure 6. Correlations in a Tailed Distribution

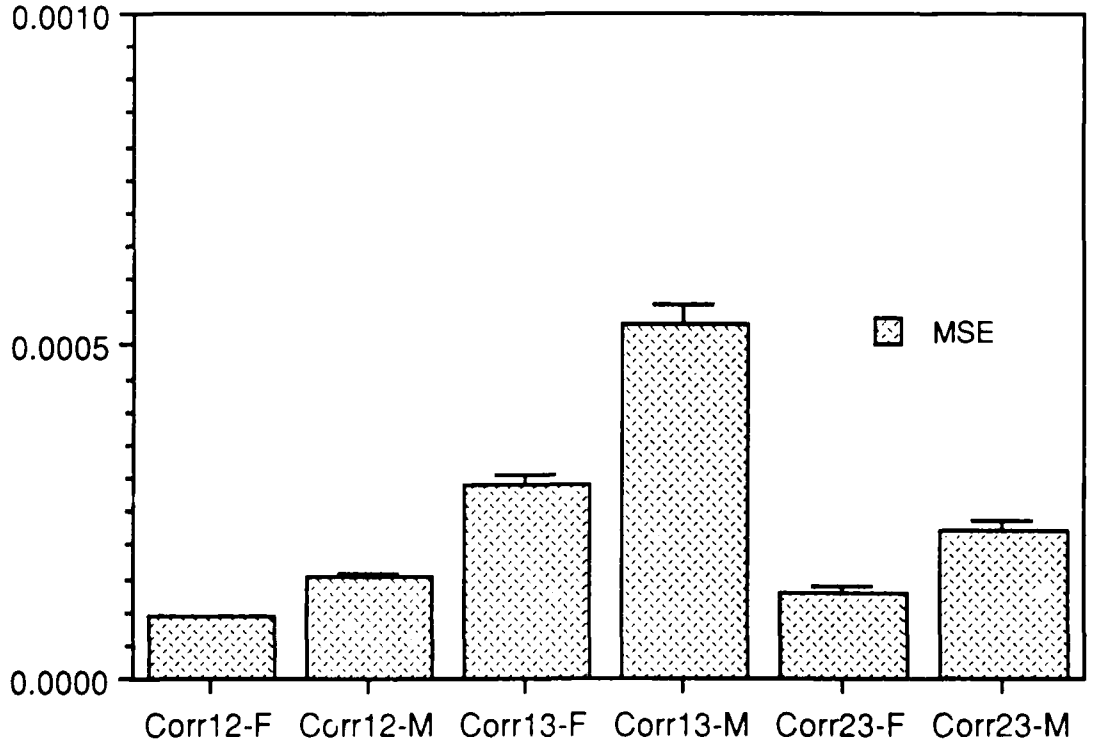


Figure 7. Bias

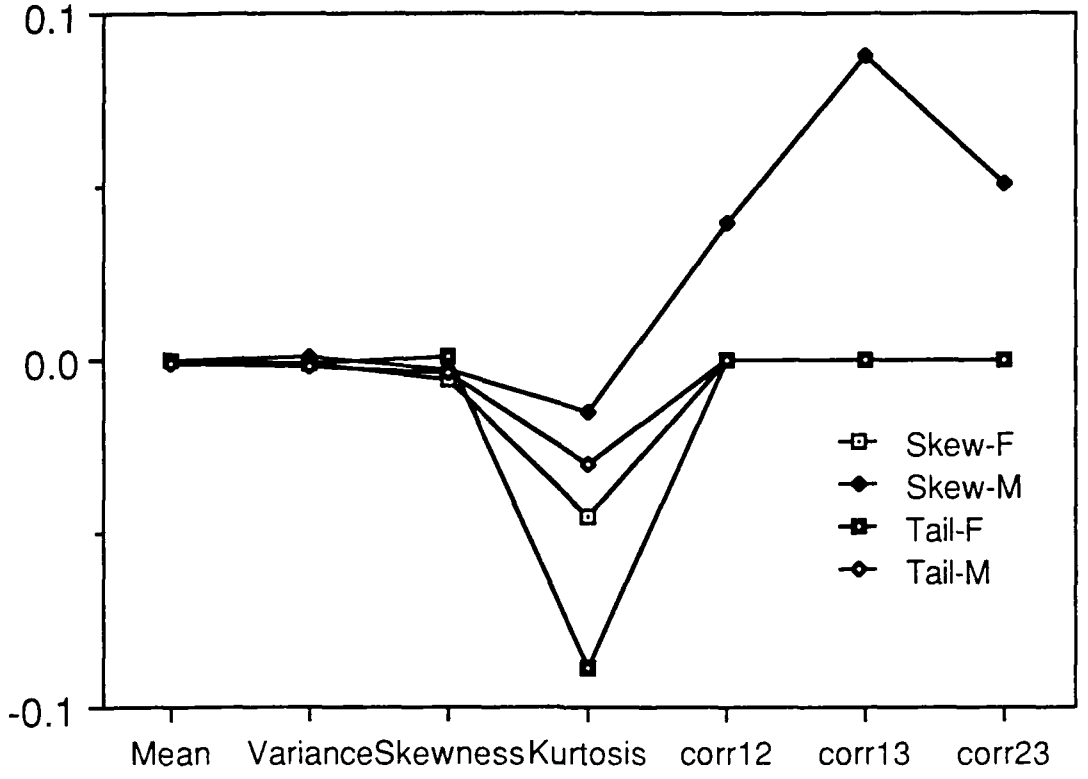


Table 2. Mean MSE and se(MSE):
Average of 50 experiments for N=2000 with 50 replications

	Simulation			
	Mix-Tail	F-Tail	Mix-Skew	F-Skew
Parameter	0.0	0.0	0.0	0.0
Mean	-0.000630	-0.000432	-0.000316	-0.000450
MSE	0.000484	0.000471	0.000495	0.000474
se(MSE)	0.000018	0.000016	0.000016	0.000016
Parameter	1.0	1.0	1.0	1.0
Var	0.997878	0.999318	0.999242	0.998956
MSE	0.002532	0.002714	0.002271	0.002218
se(MSE)	0.000062	0.000061	0.000060	0.000053
Parameter	0.0	0.0	1.062	1.062
Skew	-0.003381	0.000689	1.059118	1.056038
MSE	0.028467	0.039084	0.009048	1.131984
se(MSE)	0.000664	0.001193	0.000261	0.005053
Parameter	3.1212	3.1212	2.4366667	2.4366667
Kurtosis	3.091182	3.032055	2.421257	2.391851
MSE	0.278781	10.744024	0.119259	6.633025
se(MSE)	0.008050	0.315243	0.004014	0.148044
Parameter	0.79	0.79	0.79	0.79
corr12	0.789920	0.790241	0.829288	0.790128
MSE	0.000152	0.000092	0.001638	0.000097
se(MSE)	0.000004	0.000002	0.000015	0.000003
Parameter	0.53	0.53	0.53	0.53
corr13	0.529925	0.530370	0.617893	0.530445
MSE	0.000532	0.000289	0.008074	0.000322
se(MSE)	0.000016	0.000009	0.000058	0.000010
Parameter	0.73	0.73	0.73	0.73
corr23	0.730281	0.730270	0.780562	0.730394
MSE	0.000223	0.000130	0.002697	0.000144
se(MSE)	0.000006	0.000004	0.000025	0.000004

Note: The Cray2/4 was used for this run. The mean, variance, skewness and kurtosis for the first variate are reported. The average of 50 independent simulations, each with N=2000 repeated 50 times, is reported.

ROBUSTNESS STUDY OF SOME RANDOM VARIATE GENERATORS

Lih-Yuan Deng, Memphis State University

Abstract

Empirical study using computer-generated random numbers have been widely used where the mathematics of analyzing a statistical procedure become intractable.

There are several generating methods to produce a random sequence with the given distribution. Most, if not all, of the methods are based on the generation of independent variate from an uniform random distribution. Comparison of the different generating methods usually is done under the criterion of "efficiency". With the wide availability of a wide variety of computers, the cost of computing is reducing dramatically. Computational efficiency should not be the *only* criterion in choosing among different random number generators. We will propose a new criterion, *"robustness"*, to compare the performance of different generating schemes.

They are two basic techniques for generating variates from U(0,1): the congruential methods and feedback shift register methods. None of these is known to generate a "true" random sequence. In this paper, using beta random variate generating methods as an example, we will compare the performances of "robustness" of several generators. It is shown that some methods will perform poorly in the sense that it will quite differ from the specified distribution when the uniform generator fails "slightly".

1. Introduction

The beta family of distribution with the p.d.f. given as

$$f_X(x) = \begin{cases} \dfrac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, & 0 \le x \le 1, \\ 0, \text{elsewhere} \end{cases}$$

was used to model random processes with a finite range because of its various values of the parameters allow many shapes of the p.d.f.

There are several popular generating methods for arbitrary value of the parameters. Based on the rejection method, Jöhnk(1964) first develop a generating method for arbitrary a, b. Ahrens and Dieter(1974) proposed a more efficient generating method when both a, b are greater than one. Other methods will be more efficient under some special type of the parameter values, see Atkinson and Whittaker(1976). Cheng(1978) compared these and other method for generating beta variate based on the criterion of efficiency.

Note that all the methods are based on the successful generation of independent variate from an uniform random distribution. i.e. The theoretical distribution of the variates generated by each method will follow the beta distribution with the given parameters a, b, *if* one can generate a truly uniform numbers. They are two basic techniques are widely used: the congruential methods and feedback shift register methods. None of these is known to generate a "true" random sequence. In fact, an uniform random number generator passed a sequence of statistical test $T_1, T_2, ...T_n$, there is no guarantee that it will pass a further test T_{n+1}. In practice, an uniform random number generator will be considered "random" if several statistical tests has be passed. Another problem of an uniform random number generator is that it may sometimes display some *locally* non-random behavior , i.e. a block of numbers toward some bias, whereas next block toward the opposite bias. For further discussion, see Knuth(1969) and Kennedy and Gentle(1980).

In this paper we are concerned with the quality of a given random number generator under the situation that we failed to generate a truly uniform random numbers. Note that if a truly uniform random numbers can be generated, then all proposed methods will yield the desired distribution. And the only criterion to compare generating methods usually is "efficiency". As we shall see in Section 3, that the resulting distributions may be quite different under the "alternative" distributions. In the next section, we will consider two simple generating methods of a special beta distribution, with $a = 1, b = n$.

2. Comparisons of Generating Methods

It is easy to see that the following will generate a random variate with distribution beta(1,n):

$$X = \max(U_1, U_2, ..., U_n) \qquad \text{(A)}$$

and

$$Y = U^{\frac{1}{n}} \qquad \text{(B)}$$

where $U_1, U_2, ..., U_n$ i. i. d. $\sim U(0,1)$ and $U \sim U(0,1)$.

One can easily see that

(1) When n is small, method (A) will be more efficient than (B) because n-th root computation will take a longer computing time than sorting a small array of numbers.

(2) When n is large, method (A) will be less efficient than (B) because n-th root computation will take a shorter computing time than sorting a large array of numbers. The computing time of method (A) is known to be proportional to $n \log n$, where method (B) computing time is independent of the size n.

(3) Another advantage of method (B) over method (A) is that it can still be used even when n is non-integer, where method (A) will fail for non-integer n.

In the next section, we will consider the question of "robustness" of these two generators. That is, if U_i's do not follow an uniform distribution, then which generators will produce random variates with distribution closer to beta(1,n) distribution?

3. Robustness of Generating Methods

Note that the distribution of X and Y will follow a beta(n,1) distribution if $U_1, U_2, ...U_n$ and U in (A) and (B) follow an uniform distribution over [0,1]. We will assume that

$$U_1, U_2, ..., U_n \text{ i. i. d. } \sim F_\theta(t)$$

and

$$U \sim F_\theta(t)$$

where $F_\theta(t)$ is the c.d.f. of a distribution over [0,1] which is close to but not exactly the uniform distribution, where θ can be considered as parameter of the family of "neighborhood" distributions around uniform distribution. Without loss of generality, we assume when $\theta = 0$, $F_0(t)$ is the c.d.f. of the uniform distribution. One can easily derive the cumulative distribution function(c.d.f.) as following:

$$F_X(t) = Pr(\max(U_1, U_2, ..., U_n) \le t)$$
$$\prod_{i=1}^{n} Pr(U_i \le t) \quad [F_\theta(t)]^n \qquad (1)$$

and

$$F_Y(t) = Pr(U^{\frac{1}{n}} \le t) = F_\theta(t^n). \quad (2)$$

The c.d.f. of $Z \sim$ beta(n,1) is given as

$$F_Z(t) = t^n. \quad (3)$$

To study the relationship among $F_X(t)$, $F_Y(t)$ and $F_Z(t)$, we will make some assumptions about $F_\theta(t)$. Let $f_\theta(t)$ be the p.d.f. of $F_\theta(t)$ and for $0 \le t \le 1$,

$$f_\theta(t) = 1 + g_\theta(t). \quad (4)$$

Note that $g_\theta(t)$ represents the "deviation" of $f_\theta(t)$ from the p.d.f. of the uniform distribution. We will assume that $g_\theta(t)$ is bounded and continuous function both in t and in θ. Denote the maximum positive and negative deviation of $f_\theta(t)$ as

$$\epsilon_\theta^+ = \max_{0 \le t \le 1} g_\theta(t) \quad (5)$$

and

$$-\epsilon_\theta^- = \min_{0 \le t \le 1} g_\theta(t). \quad (6)$$

The c.d.f. $F_\theta(t)$ can be written as

$$F_\theta(t) = t + G_\theta(t), \quad 0 \le t \le 1, \quad (7)$$

where

$$G_\theta(t) = \int_0^t g_\theta(u)du. \quad (8)$$

It is easy to show that

LEMMA 1. *$\epsilon_\theta^+ \ge 0, \epsilon_\theta^- \ge 0$ and $(1-\epsilon_\theta^-) \le f_\theta(t) \le (1+\epsilon_\theta^+)$.*

PROOF: Using (7), we can see that $G_\theta(0) = G_\theta(1) = 0$. From the Mean Value Theorem, we have

$$0 = G_\theta(1) - G_\theta(0) = (1-0)g_\theta(\lambda) \quad \text{for some } \lambda \in [0,1]$$

Therefore, we have shown that

$$-\epsilon_\theta^- = \min_{0 \le t \le 1} g_\theta(t) \le 0 = g_\theta(\lambda) \le \max_{0 \le t \le 1} g_\theta(t) = \epsilon_\theta^+.$$

Lemma 1 follows from (4)-(6). ∎

The relationship among $F_X(t)$, $F_Y(t)$ and $F_Z(t)$ is summarized as in the following theorem:

THEOREM 1. *For $0 < t < 1$,*

$$(1-\epsilon_\theta^-)^n \le \frac{F_X(t)}{F_Z(t)} \le (1+\epsilon_\theta^+)^n \quad (9)$$

and

$$(1-\epsilon_\theta^-) \le \frac{F_Y(t)}{F_Z(t)} \le (1+\epsilon_\theta^+). \quad (10)$$

PROOF: From (1)-(3), we have, for $0 < t < 1$,

$$\frac{F_X(t)}{F_Z(t)} = \left[\frac{F_\theta(t)}{t}\right]^n$$

and

$$\frac{F_Y(t)}{F_Z(t)} = \frac{F_\theta(t^n)}{t^n}.$$

From (7), we have, for $0 < t < 1$,

$$\frac{F_\theta(t)}{t} = 1 + \frac{G_\theta(t)}{t}. \quad (11)$$

Applying the Mean Value Theorem, we have

$$G_\theta(t) = G_\theta(t) - G_\theta(0) = (t-0)g_\theta(\lambda), \text{ for some } \lambda \in [0,t]. \quad (12)$$

Plug (12) in (11), we get

$$\frac{F_\theta(t)}{t} = 1 + g_\theta(\lambda), \text{ for some } \lambda \in [0,t].$$

and therefore

$$(1-\epsilon_\theta^-) \le \frac{F_\theta(t)}{t} \le (1+\epsilon_\theta^+). \quad (13)$$

Inequalities in (9), (10) follows easily from (13). ∎

Theorem 1 shows that $\frac{F_Y(t)}{F_Z(t)}$ has a tighter bound than $\frac{F_X(t)}{F_Z(t)}$. Therefore $F_Y(t)$ can be much closer to $F_Z(t)$ than $F_X(t)$ to $F_Z(t)$ and the difference will be more dramatic when n is large. We will show in our next theorem that the same conclusion holds true when comparing their corresponding p.d.f.'s.

Let $f_X(t), f_Y(t)$ and $f_Z(t)$ be the p.d.f of $F_X(t), F_Y(t)$ and $F_Z(t)$, respectively. Taking the derivatives from their c.d.f.'s in (1)-(3), they can be written as the following: (for $0 \le t \le 1$)

$$f_X(t) = n[F_\theta(t)]^{n-1} f_\theta(t), \quad (14)$$

$$f_Y(t) = n f_\theta(t^n) t^{n-1} \quad (15)$$

and

$$f_Z(t) = n t^{n-1}. \quad (16)$$

The relationship among $f_X(t), f_Y(t)$ and $f_Z(t)$ is summarized as in the following theorem:

THEOREM 2. *For $0 < t < 1$, we have*

$$(1-\epsilon_\theta^-)^n \le \frac{f_X(t)}{f_Z(t)} \le (1+\epsilon_\theta^+)^n \quad (17)$$

and

$$(1-\epsilon_\theta^-) \le \frac{f_Y(t)}{f_Z(t)} \le (1+\epsilon_\theta^+). \quad (18)$$

PROOF: Form (14), (15) and (16), we can see that

$$\frac{f_X(t)}{f_Z(t)} = \left[\frac{F_\theta(t)}{t}\right]^{n-1} f_\theta(t) \quad (19)$$

and

$$\frac{f_Y(t)}{f_Z(t)} = f_\theta(t^n). \quad (20)$$

From Lemma 1, we know that

$$(1-\epsilon_\theta^-) \le f_\theta(t), f_\theta(t^n) \le (1+\epsilon_\theta^+). \quad (21)$$

Theorem 2 follows easily from (19)-(21). ∎

Theorem 2 again shows that $\frac{f_X(t)}{f_Z(t)}$ has a tighter bound than $\frac{f_X(t)}{f_Z(t)}$. Therefore $f_Y(t)$ can be much closer to $f_Z(t)$ than $f_X(t)$ to $f_Z(t)$ when the true distribution of the "uniform generator" is not really uniform, especially when n is large. Theorems 1 and 2 show that method (B) is more "robust" than method (A). One intuitive argument that (A) is not as robust as (B) is because X will follow beta(1,n) only when *each* of U_i i. i. d. $U(0,1)$, whereas $Y \sim$ beta(1,n) whenever $U \sim U(0,1)$.

An extensive empirical study of comparing the robustness of beta random number generators as well as other random number generators has been under investigation and will be reported elsewhere.

4. Summary

We have shown that if the true distribution of the so-called "uniform random generator" is slightly different from U(0,1), then various generators may yield quite different distribution than the one we try to generate. With the wide availability of the cheaper and faster computers, one should not be concerned mainly with the cost of computing time. That is, the efficiency should no longer be the only criterion to compare the performance of the generators. We propose in this paper to adopt a new criterion like "robustness" to compare the performance of different generating schemes.

REFERENCES

Ahrens, J. H. and U. Dieter (1974), "Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions," *Computing* **12**, 223-246.

Atkinson, A. C., and J. Whittaker (1976), "A Switching Algorithm for the Generation of Beta Random Variables with at Least One Parameter Less than 1," *Journal of the Royal Statistical Society(A)* **139**, 462-467.

Cheng, R. C. H. (1978), "Generating Beta Variables with Non-Integral Shape Parameter," *Communications of the ACM* **21**, 317-322.

Jöhnk, M. D. (1964), "Erzeugung von Betaverteilter und Gammaverteilter Zufallszahlen," *Metrika* **8**, 5-15.

Kennedy, W. J. Jr, and Gentle, J E. (1980), *Statistical Computing*, Marcel Dekker : New York, NY.

Knuth, D. E. (1969), *The Art of Computer Programming*, Vol 2: Seminumerical Algorithms, Addison-Wesley : Reading, Mass.

A RATIO-OF-UNIFORMS METHOD FOR GENERATING EXPONENTIAL POWER VARIATES

Dean M. Young, Baylor University
Danny W. Turner, Baylor University
John W. Seaman, Jr., University of Southwestern Louisiana

1. Introduction

The standardized exponential power distribution (EPD) family has probability density function

$$g_\alpha(x) = \frac{1}{2\Gamma(1 + 1/\alpha)} \exp(-|x|^\alpha), \quad -\infty < x < \infty, \quad \alpha \geq 1.$$

This family is symmetric about zero and contains members with a variety of tail shapes from the uniform ($\alpha \to \infty$) to the normal ($\alpha = 2$) to the double exponential ($\alpha = 1$). Because of the diversity of available tail shapes, the EPD family has proven useful in robustness studies. For a review of such applications and others, see Box and Tiao (1973) and Tadikamalla (1980).

Johnson (1979) and Johnson, Tietjen, and Beckman (1980) have provided direct transformation methods for EPD random variate generation. Tadikamalla (1980) has derived generalized rejection techniques for EPD random variate generation. He has provided two algorithms, called ED for $1 \leq \alpha < 2$ and EN for $2 \leq \alpha \leq 6$. (Values of α greater than 6 are of little interest because of their extreme kurtosis values.) Tadikamalla has found the combination of ED and EN, hereafter referred to as ED/EN, to be superior to the gamma transformation methods for $1 \leq \alpha \leq 6$. For convenience we present ED and EN below. See Tadikamalla (1980) for more discussion of these algorithms. We denote the uniform distribution over a set S by US. We denote the normal distribution with mean m and variance v by N(m,v).

Algorithm ED: (for $1 \leq \alpha < 2$)

Step 0. Compute $A = 1/\alpha$, $B = A^A$
(Required once for each α).

Step 1. Generate a double-exponential variate X:
a) Generate U from U(0,1).
b) If $U > .5$, then $X = B(-\ln(2(1-U)))$.
Otherwise, $X = B\ln(2U)$.

Step 2. Generate R from U(0,1).

Step 3. Test of acceptance/rejection: If $\ln(R) > (-|X|^\alpha + |X|/B - 1 + A)$, then go to Step 1. Otherwise, return X.

Algorithm EN: (for $2 \leq \alpha$)

Step 0. Compute $A = 1/\alpha$, $B = A^A$.
(Required once for each α).

Step 1. Generate X from $N(0,B^2)$.

Step 2. Generate R from U(0,1).

Step 3. Test of acceptance/rejection: If $\ln(R) > (-|X|^\alpha + X^2/2B^2 + A - .5)$, then go to Step 1. Otherwise, return X.

In this paper we develop a simpler, ratio-of-uniforms method of EPD random variate generation and compare it to Tadikamalla's ED and EN algorithms for $1 \leq \alpha \leq 6$. It is found that generation times for the ratio-of-uniforms method are uniformly better than ED and EN in this range.

2. The Ratio-of-Uniforms Method for EPD Variate Generation

In this section we shall briefly review the ratio-of-uniforms (ROU) method and apply it to EPD variate generation. For a more thorough review of the ROU method, see, for example, Devroye (1986).

First proposed by Kinderman and Monahan (1977), the ROU method has been studied by several authors, including Kinderman and Monahan (1979), Cheng and Feast (1979), Robertson and Walls (1980), and Barbu (1983). The method is based on the following result, due to Kinderman and Monahan (1977).

Theorem 2.1 Suppose f is an nonnegative integrable function on the real numbers. Let the random vector (U,V) have a uniform distribution over the set

$$D = \{(u,v): 0 \leq u \leq \sqrt{f(v/u)}\}.$$

Then, V/U has density f/c, where $c = 2[\text{area}(D)]$.

The basic idea is to enclose D in some simple set E, generate observations from a uniform distribution on E, and apply the rejection principle. The following result is proved in Devroye (1986, p. 195).

Theorem 2.2 Let f and D be defined as above. Let b, a_-, and a_+ be constants such that

$$b \geq \sup_u \sqrt{f(u)},$$

$$a_- \leq \inf_u u\sqrt{f(u)}, \text{ and}$$

$$a_+ \geq \sup_u u\sqrt{f(u)}.$$

Let E be the rectangle formed by the Cartesian product $[0,b) \times [a_-,a_+]$. Then, the set D can be enclosed in E if and only if f(u) and $u^2 f(u)$ are bounded for all u. With these theoretical results in place, we now present the general ROU algorithm. Again, we denote the uniform distribution over a set S by US.

Algorithm ROU:

Step 0. Compute b, a_-, and a_+ (required once for each α).

Step 1. Generate U from U[0,b).

Step 2. Generate V from $U[a_-, a_+]$.

Step 3. Set $X = V/U$.

Step 4. If $U^2 \le f(X)$, then return X. Otherwise, go to Step 1.

For the EPD application, we take $f_\alpha(x) = \exp(-|x|^\alpha)$. It is easy to show that $b = 1$, $a_+ = (2/e\alpha)^{1/\alpha}$, and $a_- = -(2/e\alpha)^{1/\alpha}$ satisfy the conditions of Theorem 2.2. Note that only one calculation is required in Step 0 since $a_- = -a_+$ and b is constant for all α.

3. Comparison of the Algorithms

We begin by considering ease of implementation. Set-up times for ROU and ED/EN are comparable, requiring the generation of one constant involving a single exponentiation (Step 0 in each algorithm). However, in simulations involving more than one α value, one must decide whether to use ED or EN according to the value of α so that set-up for ED/EN is slightly more complicated.

Algorithm ROU requires the generation of two uniform random variates (Steps 1 and 2) which must be combined in a ratio (Step 3). One comparison is made (Step 4) involving an evaluation of the function $f_\alpha(x)$. In contrast, an application of ED requires the generation of two uniform random variates (Steps 1 and 2), both of which must be evaluated in a logarithmic function (Steps 1 and 3). Furthermore, two comparisons are required (Steps 1 and 3), one of which (Step 3) requires the evaluation of a function more complicated than $f_\alpha(x)$. Algorithm EN is similarly complex but involves the calculation of a uniform and a normal deviate rather than of two uniforms. Clearly, ROU is much simpler than ED/EN. Devroye (1986, p. 8) has noted that among factors that play an important roles in the choice of random variate generators "[simplicity and readability are] perhaps the most neglected in the literature." Algorithm ROU should certainly be selected over ED/EN on the basis of these criteria.

We now consider efficiency of the algorithms. Devroye (1986, p. 196) has derived the expected number of iterations per variate produced--the so-called rejection constant--for the general ROU algorithm. Its general form is given below:

$$r_{ROU} = \frac{b(a_+ - a_-)}{\text{area}(R)} = \frac{2b(a_+ - a_-)}{\int_{-\infty}^{\infty} f(u)\,du}$$

Written as a function of α, the rejection constant for the EPD family is given by

$$r_{ROU}(\alpha) = \frac{2(2/\alpha e)^{1/\alpha}}{\Gamma(1 + 1/\alpha)}, \quad 0 \le \alpha < \infty.$$

Tadikamalla (1980) has provided rejection constants for ED and EN. As functions of α they may be written as the following:

$$r_{ED}(\alpha) = \frac{(1/\alpha)^{1/\alpha}}{\Gamma(1 + 1/\alpha)} e^{(1 - 1/\alpha)}, \quad 1 \le \alpha < 2,$$

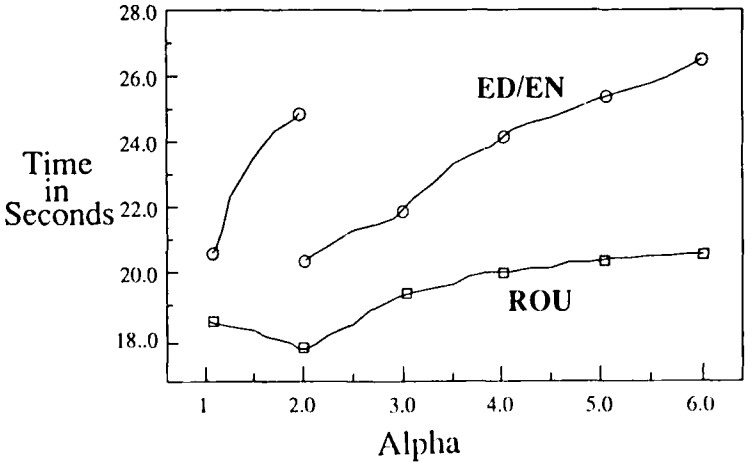
and

$$r_{EN}(\alpha) = \frac{\sqrt{\pi/2}(1/\alpha)^{1/\alpha}}{\Gamma(1 + 1/\alpha)} e^{(.5 - 1/\alpha)}, \quad \alpha \ge 2.$$

We shall use these rejection constants as measures of efficiency, where efficiency is defined as the reciprocal of the rejection constant. A graph of the efficiency functions $1/r_{ROU}$, $1/r_{ED}$, and $1/r_{EN}$, is shown in Figure I for various values of α. For the values of α that are

FIGURE I

Trimmed-Mean Time
To Generate 10,000 Observations from
Exponential Power Distributions for Varied Powers



of interest, ED/EN is uniformly more efficient than ROU. However, efficiency is a function of expected number of iterations per variate. It will be seen that the greater complexity of algorithm ED/EN requires longer time per iteration relative to algorithm ROU, thus negating the value of higher efficiency.

To compare generation speeds, we have coded the algorithms in SAS-PROC MATRIX on an IBM 4381 Model P22. Algorithm EN requires the generation of normal random variates. The SAS function RANNOR has been utilized for this purpose. The PROC MATRIX implementation of each algorithm has been used to generate 50,000 variates in five independent runs of 10,000 variates each for $1 \leq \alpha \leq 6$. Extreme generation-time values have been trimmed from each set of five runs. Figure I gives the trimmed average generation times in seconds for each algorithm. As can be seen from Figure I, algorithm ROU exhibits uniformly faster performance than algorithm ED/EN.

4. Conclusions

We have presented a ratio-of-uniforms algorithm, called ROU, for exponential random variate generation and have compared it to a generalized rejection method, called ED/EN, developed by Tadikamalla (1980). We have demonstrated that while ROU is inferior to ED/EN with respect to efficiency (iterations required per random variate), it is markedly superior in generation time, which is, practically, the most important measure of performance. Furthermore, a direct comparison of algorithms ROU and ED/EN clearly indicates that ROU is far more simple and easily implemented. Devroye (1986, p. 11) notes that, "It is a general rule in computer science that speed can be reduced by using longer, more sophisticated programs." Happily, our comparison of ROU with ED/EN seems to provide an exception to that rule.

5. Acknowledgement

The authors wish to thank Professor Luc Devroye for his helpful comments and encouragement.

References

Barbu, G. (1983). On computer generation of random variables as a ratio of uniform random variables. *Economic Computation and Economic Cybernetics Studies and Research*, Academy of Economic Studies, Bucharest, Vol. **18**, 33-50.

Box, G.E.P. and Muller, M.E. (1958). A Note on the Generation of Random Normal Deviates. *Annals of Mathematical Statistics*, **29**, 610-611.

Box, G.E.P. and Tiao, George C. (1973). *Bayesian Inference in Statistical Analysis*, Reading, Mass.: Addison-Wesley, 149-202.

Cheng, R.C.H. and Feast, G.M. (1979). Some simple gamma variate generators. *Applied Statistics*,**28**, 290-295.

Devroye, L. (1986). *Non-uniform random variate generation*. Springer-Verlag: New York.

Johnson, M.E. (1979). Computer Generation of the Exponential Power Distributions. *Journal of Statistical Computation and Simulation*, **9**, 239-240.

Johnson, M.E., Tietjen, G.L., and Beckman, R.J. (1980). A new family of probability distributions with applications to Monte Carlo studies. *Journal of the American Statistical Association*, **75**, 276-279.

Kinderman, A.J. and Monahan,J.F. (1977). Computer generation of random variables using the ratio of uniform deviates. *ACM Transactions on Mathematical Software*, **3**,257-260.

Kinderman, A.J. and Monahan,J.F. (1979). New methods for generating student's t and gamma variables. Technical Report, Department of Management Science, California State University, Northridge, CA..

Kinderman, A.J., and Ramage, J.G. (1976). Computer Generation of Normal Random Variables. *Journal of the American Statistical Association*, **71**, 893-896.

Robertson, I. and Walls, L.A. (1980). Random number generators for the normal and gamma distributions using the ratio of uniforms method. Technical Report AERE-R 10032, U.K. Atomic Energy Authority, Harwell, Oxfordshire.

Tadikamalla, P.R. (1980). Random sampling from the exponential power distribution. *Journal of the American Statistical Association*, **75**, 683-686.

AN APPROACH FOR GENERATION OF TWO VARIABLE SETS WITH A SPECIFIED CORRELATION AND FIRST AND SECOND SAMPLE MOMENTS

Mark Eakin, Ph.D. and Henry D. Crockett, C.S.P.

ABSTRACT

Certain simulations require the generation of correlated variables with a prespecified first and second moments. The first step involved the random generation of two standardized variables. Secondly, the first variable was replaced by a linear combination of the two variables such that the coefficient of the linear combination and the second variable. The variables can then be adjusted to give the required first second sample moments without modifying the correlation equations.

INTRODUCTION

This paper presents a way of generating two real-valued variables that have a fixed sample correlation. Edwards (1959) and Searle and Firey (1980) discuss procedures to generate two integer valued variables that have a specified correlation. Both procedures require several iterations in order to achieve the desired correlation. However, in large-scale simulations the iterative approach is not efficient. Kvalseth (1979) developed a procedure to generate a pair of normally distributed variables that had a specified sample correlation value.

The following procedure gives a closed-form solution to the problem of achieving a fixed sample correlation between two real valued variables. The two variables do not have to be normally distributed but may have prespecified sample means and variances.

The problem: generate two variables, x_1 and x_2, from samples of size n such that (1) the mean of x_1 and x_2 are M_1 and M_2, respectively; (2) the standard deviations are s_1 and s_2, respectively, and (3) the correlation between x_1 and x_2 is r_x.

The solution: (1) generate two variables z_1 and z_2 and standardize their values using a sample of size n; (2) calculate the correlation, r_3, between z_1 and z_2; and (3) let

$$x_1 = z_1 \; s_1 + M_1 \qquad (1)$$

and

$$x_2 = (z_1 \; c + z_2) \; s_2 + M_2 \qquad (2)$$

where $$c = (-K_2 + (K_2 2 - 4 \; K_1 \; K_3)^{1/2})/(2K_1) \qquad (3)$$

$$K_1 = r_x^2 - 1 \qquad (4)$$

$$K_2 = 2r_3 \; (r_x^2 - 1) \qquad (5)$$

and

$$K_3 = (r_x^2 - r_3^2) \qquad (6)$$

The proof consists of finding the value of c such that the correlation of z_1 and $(z_1 \; c + z_2)$ is r_x. The values of z_1 and $(z_1 \; c + z_2)$ are then adjusted to give the necessary means and standard deviations. The proof starts by expressing the square of the correlation between z_1 and $(z_1 \; c + z_2)$ in terms of sums and products and squares (usually this is expressed in terms of deviations from the mean but both z_1 and $(z_1 \; c + z_2)$ have mean zero):

$$r_x 2 = \frac{[\; z_1 \; (z_1 \; c + z_2)/(n-1)]^2}{[\; z_1 2/(n-1)] \; [\; (z_1 \; c + z_2)^2/(n-1)]} \qquad (7)$$

Multiplying the terms together in the numerator and denominator of (7) and recalling that the variance of z_1 is one gives

$$rx^2 = \frac{[\; (c \;\; z_1 + \;\; z_1 z_2)/(n-1) \;]^2}{[1] \; [(c2 \;\; z12 + c \;\; z1z2 + \;\; z22)/(n-1)]} \qquad (8)$$

The following identities will be substituted into (8):

$$4_3 = z_1 z_2/(n-1) \quad , \qquad (9)$$

$$z_1 2/(n-1) = 1 \quad , \text{ and} \qquad (10)$$

$$z_2 2/(n-1) = 1 \qquad (11)$$

obtaining

$$r_x^2 = \frac{[\; 3 + r_3 \;]^2}{[c^2 + cr_3 + 1]} \qquad (12)$$

Squaring the numerator of (12), multiplying both sides by the denominator, and then gathering all terms on the left hand side obtains

$$(c^2 + 2cr_3 + 1) \; r_x^2 - c^2 \; 2cr_3 - r_3^2 = 0. \qquad (13)$$

Rewriting as a quadratic function of a c results in

$$(r_x^2 - 1)c + 2r_3(r_x^2 - 1)c + (r_x^2 - r_3^2) = 0 \qquad (14)$$

The solution to (14) can be found using the quadratic formula for the following quadratic equation

$$K_1 \; c + K_2 \; c + K_3 = 0 \qquad (15)$$

where $K_1 = r_x^2 - 1$ (16)

$K_2 = 2r_3 (r_x^2 - 1)$ (17)

and $K_3 = (r_x^2 - r_3^2)$. (18)

Bibliography

Edwards, Bruce (1959), "Constructing Simple Correlation Problems with Pre-Determined Answers," The American Statistician, 13, 25-27.

Kvalseth, Tarald (1979), "A Simple Method of Generating Correlated Data," Perceptual and Motor Skills, 48, 891-895.

Searle, S.R. and P.A. Firey (1980), "Computer Generation of Data Sets for Homework Exercises in Simple Regression," 34, 51-54.

XIV. ROBUST AND NONPARAMETRIC METHODS

GAMMA PROCESSES, PAIRED COMPARISONS AND RANKING

Hal Stern, Harvard University

Introduction

In non-parametric statistical procedures the n observations in a sample are often replaced by their ranks within the sample. Under the null hypothesis, the distribution on the ranks is assumed to be uniform over permutations of the integers from 1 to n. Other distributions on the permutations are of interest as alternative descriptions. Mallows (1957) introduces a variety of alternative distributions. In this discussion we consider models for rank data which are derived by considering permutations of gamma random variables. It turns out that these models include the two most popular ranking models. After some discussion motivating the use of gamma random variables, the gamma models are applied to paired comparisons experiments. In these experiments only two of the set of objects are ranked at one time. This simple case leads to some theoretical results about gamma models. Finally, a data set consisting of the results of horse races is analyzed using the methods described here. The gamma models are used to model the observed distribution on permutations.

Gamma Comparison Models

Suppose that k individuals are to be ranked according to the waiting time for r points to be scored. If the i^{th} individual scores points as a Poisson process with parameter λ_i (the time between points is an exponential random variable with mean λ_i^{-1}) then the time until r points are scored has the gamma distribution with shape parameter r and scale parameter λ_i. By also assuming that the waiting times for the k individuals are independent we can compute the probability that the k individuals are ranked in any order. Let $\pi = (\pi_1, \ldots, \pi_k)$ be a permutation of the integers from 1 through k and let $X_1, \ldots, X_k$ be independent gamma random variables with shape parameter r and different scale parameters $\lambda_1, \ldots, \lambda_k$. The probability that individual π_1 is ranked first, π_2 is ranked second, etc. is given by the k-dimensional integral

$$p^{(r)}(\pi) = Pr(X_{\pi_1} < X_{\pi_2} < \cdots < X_{\pi_k}). \qquad (1)$$

This heuristic derivation is restricted to integer values of r. Other values of r can also be considered if the point scoring process is modeled as an independent increments gamma process. The gamma process is a stochastic process with parameter λ, $G_\lambda(r)$, such that $G_\lambda(0) = 0$, $G_\lambda(r_2)$-$G_\lambda(r_1)$ is independent of $G_\lambda(r_1)$-$G_\lambda(r_0)$ whenever $r_0 \le r_1 \le r_2$ and $G_\lambda(r_2) - G_\lambda(r_1)$ has the gamma distribution with shape parameter $r_2 - r_1$ and scale parameter λ. Then the probability of a particular permutation is defined for all positive values of r.

Paired Comparisons

In many experimental situations it is not reasonable to rank more than two objects at a time. In ranking tennis or chess players the only observations are the results of matches between two players. From these results, we hope to rank all of the players. This is an example of a paired comparison experiment. A bibliography of the paired comparison literature is provided by Davidson and Farquhar (1976).

Suppose that the k players or objects to be compared are identified by the numbers $1, \ldots, k$. The probability that i is preferred to j in a comparison is denoted $p_{ij}^{(r)}$. Conceptually $p_{ij}^{(r)}$ is the marginal probability that i is ranked before j in the distribution $p^{(r)}(\pi)$. It can also be derived by considering a comparison of two independent gamma random variables with shape parameter r and differing scale parameters. In this case

$$\begin{aligned} p_{ij}^{(r)} &= Pr(X_i < X_j) \\ &= \int_0^\infty \int_0^{x_j} \frac{\lambda_i^r x_i^{r-1} e^{-\lambda_i x_i} \lambda_j^r x_j^{r-1} e^{-\lambda_j x_j}}{\Gamma(r)\Gamma(r)} dx_i dx_j \qquad (2) \\ &= \int_0^\infty \int_0^{\frac{\lambda_i}{\lambda_j} x_j} \frac{z_i^{r-1} e^{-z_i} z_j^{r-1} e^{-z_j}}{\Gamma(r)\Gamma(r)} dz_i dz_j \\ &= f_r(\lambda_i/\lambda_j) \end{aligned}$$

For fixed r, the probability that i defeats j is increasing in the ratio λ_i/λ_j. This is consistent with the interpretation of λ_i as the rate at which points are scored by player i. It is also true that for fixed ratio λ_i/λ_j greater than one, the probability that i defeats j is increasing in the parameter r which measures the length of the game. More complicated models can be developed to take into account covariate information or the possibility of ties.

Examples

The parameter r determines the shape of the gamma variables to be compared. By considering specific values of r some natural models are obtained. If r is equal to one then the probability that i is preferred to j is $\lambda_i/(\lambda_i + \lambda_j)$. This is the Bradley-Terry paired comparison model (Bradley and Terry (1952), Bradley (1953, 1954, 1955)). The Bradley-Terry model has a long history of derivations and interpretations including Zermelo (1929) and Ford (1957). Of the many alternative derivations it is important to mention the convolution type linear model approach discussed by David (1963), Latta (1979) and Bradley (1953). If player i's score has the extreme value distribution with location parameter $\ln \lambda_i$ and player j's score has the extreme value distribution with location parameter $\ln \lambda_j$ then the probability that i defeats j is given by the Bradley-Terry model. It turns out that for any value of r there is a convolution type linear model which is equivalent to the gamma paired comparison model (Stern 1987). For other values of r the extreme value distribution is replaced by a different translation family of densities.

Other integer values of r can be easily interpreted in terms of the Poisson point scoring model. When r equals two, the probability that i defeats j is the probability that player i scores two points before player j does. This can be computed directly from gamma random variables with shape parameter 2 or indirectly as a sequence of comparisons with $r=1$. Noninteger values may also be considered. They are included in this discussion by virtue of the independent increments Gamma process described earlier.

When large values of r are considered, the gamma paired comparison model tends to the Thurstone-Mosteller model (Thurstone 1927, Mosteller 1951). Thurstone (1927) assumes that comparisons between two objects are determined by comparisons of two normally distributed random variables. Five different models are

derived by making different assumptions about the joint distribution of the normal random variables. Mosteller (1951) discusses various properties of Thurstone's model V in which the normal random variables are assumed to have equal variances. The distribution of the standardized gamma random variable with shape parameter r and scale parameter λ tends to a standard normal distribution as r gets large. Thus comparisons between gamma random variables lead to the Thurstone-Mosteller model for large values of r. The gamma model is again found to be equivalent to a convolution type linear model. More details of the relationship between the Thurstone-Mosteller model and the gamma model with large r are found in Stern (1987).

As a last special case, consider comparisons of gamma random variables with shape parameter near zero. The distribution of the logarithm of such a gamma random variable tends to the exponential distribution. Thus a paired comparison between two such gamma random variables is equivalent to a comparison of two exponential random variables with different location parameters.

Inferences in Paired Comparisons

Given the results of a series of comparisons involving k objects, the statistical experimenter would like to predict future comparisons or find the optimal ranking of the k objects. In an experiment using gamma random variables with shape parameter r and scale parameters $\lambda_1, \ldots, \lambda_k$, estimates of the parameters are required and goodness of fit tests can then be used to determine whether the model is appropriate. The usual formulation of the experiment treats the series of comparisons as independent binomial trials. If r and $\lambda_1, \ldots, \lambda_k$ are considered fixed then the probability that object i is preferred to object j is $p_{ij}^{(r)} = f_r(\lambda_i/\lambda_j)$. Let n_{ij} be the number of comparisons of objects i and j and let a_{ij} be the number of times that i is preferred to j. Then the likelihood is

$$\prod_{i=1}^{k} \prod_{j>i} \binom{n_{ij}}{a_{ij}} f_r(\lambda_i/\lambda_j)^{a_{ij}} f_r(\lambda_j/\lambda_i)^{a_{ji}}. \qquad (3)$$

The shape parameter r may be considered fixed or treated as a parameter to be estimated. In the former case r determines the nature of the comparison and may be chosen before the experiment is carried out. The maximum likelihood estimates for $\lambda_1, \ldots, \lambda_k$ may be determined using ordinary numerical algorithms when r is known. Ford (1957) describes a procedure when r is equal to one. The asymptotic normality of the maximum likelihood estimates follows from the usual maximum likelihood theory (Lehman 1983). The calculation of estimates is more complicated when r is treated as a parameter to be estimated. Typically several values of r are considered and the r which achieves the largest maximum value for the likelihood is the estimate. Prior beliefs can be incorporated in a Bayesian analysis (Davidson and Solomon (1973), Leonard (1977), Stern (1987)).

In applications it is often desired to compare the fit of a variety of models. Here we would like to compare the fit for several values of the shape parameter r. The usual log-likelihood ratio statistic is

$$Q = \sum_{i=1}^{k} \sum_{j \neq i} a_{ij} \ln \frac{a_{ij}/n_{ij}}{f_r(\hat{\lambda}_i/\hat{\lambda}_j)} \qquad (4)$$

where $\hat{\lambda}_1, \ldots, \hat{\lambda}_k$ are the maximum likelihood estimates. For large samples Q has a chi-square distribution with $(k-1)(k-2)/2$ degrees of freedom. An alternative goodness of fit procedure is described by Mosteller (1951).

Applications to Data

In the 1986 National League baseball season each National League baseball team played between eleven and eighteen games against each of the other eleven teams. The results are stored in the following matrix

$$\begin{pmatrix}
- & 8 & 12 & 10 & 12 & 17 & 7 & 8 & 7 & 10 & 9 & 8 \\
10 & - & 6 & 10 & 8 & 11 & 6 & 5 & 9 & 6 & 7 & 8 \\
6 & 12 & - & 9 & 7 & 11 & 5 & 5 & 7 & 7 & 4 & 6 \\
8 & 8 & 9 & - & 10 & 11 & 4 & 5 & 5 & 4 & 7 & 7 \\
6 & 9 & 10 & 8 & - & 7 & 4 & 5 & 6 & 6 & 6 & 3 \\
1 & 7 & 7 & 7 & 11 & - & 6 & 2 & 4 & 8 & 4 & 7 \\
5 & 6 & 7 & 8 & 8 & 6 & - & 14 & 9 & 10 & 10 & 13 \\
4 & 7 & 7 & 7 & 7 & 10 & 4 & - & 9 & 9 & 10 & 12 \\
5 & 3 & 5 & 7 & 6 & 8 & 9 & 9 & - & 10 & 10 & 11 \\
2 & 6 & 5 & 8 & 6 & 4 & 8 & 9 & 8 & - & 12 & 6 \\
3 & 5 & 8 & 5 & 6 & 8 & 8 & 8 & 8 & 6 & - & 8 \\
4 & 4 & 6 & 4 & 9 & 5 & 5 & 6 & 7 & 12 & 10 & -
\end{pmatrix}$$

where the i, j^{th} element is the number of times that team i defeated team j. The gamma paired comparison model was fit to this data (Stern 1987) with r=0.1, 0.5, 1,2,3,5,10, 20. In each case the goodness of fit statistic indicates an adequate fit. Also, the predicted results $\hat{a}_{ij} = n_{ij} f_r(\hat{\lambda}_i/\hat{\lambda}_j)$ differ by at most 0.1 over the range of r's considered. These results are consistent with the results obtained from other baseball, basketball and football seasons. Simulations also indicate that models with different values of r lead to similar fits unless the sample size is extremely large. This is consistent with the observations of Latta (1979), Burke and Zinnes (1965), and Jackson and Fleckenstein (1957). Each of these authors found that the Bradley-Terry model (r=1) and the Thurstone-Mosteller model (r large) lead to similar fits for a given data set.

Why do all paired comparisons models lead to similar fits for moderate sample sizes? The answer to this question can be determined using the triples function or composition rule of a model. The composition rule is the formula that is used to determine p_{ik}, the probability that i beats k, from p_{ij} and p_{jk}. Each model has a different value p_{ik} for a particular pair of values of p_{ij} and p_{jk}. However it turns out that the different values are quite close. Thus large numbers of comparisons of i and k are required in order to distinguish between the models. Simple calculations (Stern 1987) indicate that 500 or more comparisons of each pair of teams are required.

Ranking

Given the similarity of the gamma paired comparison models for a wide range of r, it is natural to wonder if the models corresponding to different values of r have different properties in the general ranking problem. Recall that the gamma model with parameter r for permutations of k objects is obtained by taking the probability of the permutation π to be equal to the probability that k independent gamma random variables with shape parameter r and different scale parameters are ranked according to the permutation π. The probability of the permutation $\pi = (\pi_1, \ldots, \pi_k)$, in which π_i is the index of

the object with rank i, is denoted by $p^{(r)}(\pi)$. Marginals of this distribution are represented by identifying the particular event whose probability is being described. Thus we write $p^{(r)}(i)$ for the probability that object i is ranked first ($\pi_1 = i$) and $p^{(r)}(ij)$ for the probability that i is ranked first and j is ranked second. We can take $\pi^{-1}i$ to be the rank of object i, that is to say $\pi^{-1}i = j$ if and only if $\pi_j = i$ or equivalently object i has rank j. Then $p^{(r)}(\pi^{-1}i < \pi^{-1}j)$ is the probability that i is ranked ahead of j.

Again the case r equal to one corresponds to the most commonly used ranking model. It represents the natural generalization of the Bradley-Terry model to more than two objects (Bradley, 1965). Let $\lambda_1, ..., \lambda_k$ be the scale parameters associated with the k objects. Then $p^{(1)}(\pi) =$

$$\frac{\lambda_{\pi_1}}{\sum_{i=1}^{k}\lambda_{\pi_i}}\frac{\lambda_{\pi_2}}{\sum_{i=2}^{k}\lambda_{\pi_i}}\frac{\lambda_{\pi_3}}{\sum_{i=3}^{k}\lambda_{\pi_i}}\cdots\frac{\lambda_{\pi_{k-1}}}{\sum_{i=k-1}^{k}\lambda_{\pi_i}}\frac{\lambda_{\pi_k}}{\lambda_{\pi_k}}. \qquad (5)$$

This formula has a natural interpretation in terms of a sequential ranking procedure. The probability that object π_1 is ranked first according to the gamma model with shape parameter one is equal to the probability that an exponential random variable with parameter λ_{π_1} (mean $\lambda_{\pi_1}^{-1}$) is the smallest of k exponentials with parameters $\lambda_1, ...\lambda_k$. This is precisely the first factor of $p^{(1)}(\pi)$. The second factor is the probability that object π_2 is ranked first in a comparison of the remaining k-1 objects (those not ranked first). The marginal probabilities are easy to specify due to this property. For example

$$p^{(1)}(i) = \frac{\lambda_i}{\sum_{m=1}^{k}\lambda_m}$$

and

$$p^{(1)}(ij) = \frac{\lambda_i\lambda_j}{\sum_{m=1}^{k}\lambda_m(\sum_{m=1}^{k}\lambda_m - \lambda_i)}.$$

Henery (1981) discusses the derivation of (5) using exponential random variables as we have described here. This model also has the property that the conditional probability of some events can be written in the same form as $p^{(r)}(\pi)$. Suppose that we condition on the event that object i is ranked first, then

$$p^{(1)}(\pi|\pi_1 = i) = \frac{p^{(1)}(\pi)}{p^{(1)}(i)} = \frac{\lambda_{\pi_2}}{\sum_{i=2}^{k}\lambda_{\pi_i}}\cdots\frac{\lambda_{\pi_k}}{\lambda_{\pi_k}} \qquad (6)$$

is precisely the probability that the k-1 remaining objects are ranked according to the permutation $\bar{\pi} = (\pi_2, \cdots, \pi_k)$. Harville (1973) proposed the formula (5) based on this property.

Gamma models for distributions of permutations with shape parameters other than one are difficult to work with because there are no simple expressions for $p^{(r)}(\pi)$. The cases in which r tends to ∞ and r tends to zero can be analyzed by considering the equivalent translation family models. Thus as r tends to infinity the gamma model resembles the extension of the Thurstone-Mosteller model proposed by Daniels (1950).

As a last example of gamma ranking models we consider integer values of r greater than one, particularly r equal to two. The probability of the permutation π under the gamma model with integer shape parameter r can be calculated using a counting argument. To describe this argument suppose that there are k players each attempting to score r points. Each player scores points as a Poisson process and the k processes are independent. Player i scores points at rate λ_i. At first all k players are attempting to score points simultaneously. This can be viewed as a combined Poisson process with rate $\sum_{i=1}^{k}\lambda_i$. The probability that a point in the combined process is scored by player i is proportional to λ_i. Successive points are scored independently due to the Poisson processes involved. When the first player has accumulated r points, the corresponding process is removed from the combined process. Now k-1 players compete simultaneously. This counting argument leads to a complicated expression for $p^{(r)}(\pi)$ (Stern 1987). A similar expression is obtained by Henery (1983). As a particular example if k=3 and r=2 then $p^{(2)}(\pi) =$

$$\lambda_{\pi_1}^2\lambda_{\pi_2}^2\left(\frac{6\lambda_{\pi_3}+2}{\lambda_{\pi_2\pi_3}} + \frac{4\lambda_{\pi_3}+1}{\lambda_{\pi_2\pi_3}^2} + \frac{2\lambda_{\pi_3}}{\lambda_{\pi_2\pi_3}^3}\right) \qquad (7)$$

where $\lambda_{\pi_2\pi_3} = \lambda_{\pi_2} + \lambda_{\pi_3}$ and without loss of generality we have set $\lambda_{\pi_1} + \lambda_{\pi_2} + \lambda_{\pi_3} = 1$. The justification for introducing this constraint appears in the next section. These models no longer have the sequential or conditional properties described previously for the model with r equal to one. However, applications are given later in which they lead to better fits.

Inference in Ranking Models

In a sample of n random permutations we denote the empirical distribution by $p_n(\pi)$. The log likelihood under the gamma model with shape parameter r is

$$\sum_{\pi} np_n(\pi)\ln p^{(r)}(\pi) + C \qquad (8)$$

where C is a constant that does not depend on r or the parameters $\lambda_1, \ldots, \lambda_k$. For large k it is usually not feasible to record the entire empirical distribution. Instead several marginals of the empirical distribution are recorded. For example we may only know $p_n(1)$, $\ldots, p_n(k)$, the frequencies with which each object is ranked first in the data set. Estimates of $\lambda_1, \ldots, \lambda_k$ are computed using maximum likelihood methods. Often the E-M algorithm (Dempster, Laird, Rubin (1977)) can be used.

Maximum likelihood estimation is particularly straightforward when the empirical distribution is known for k marginals which are mutually exclusive and which exhaust the possible permutations. The example cited above would be such a case. In this case the likelihood equations reduce to a particularly simple form. Estimates are obtained by solving the system of equations obtained by setting each empirical marginal probability equal to the marginal probability expressed in terms of the parameters. We consider two examples which are then used in an application.

In both examples, the empirical probabilities $p_n(1), \ldots, p_n(k)$ are assumed known. For the gamma model with shape parameter one the maximum likelihood estimates for $\lambda_1, \cdots \lambda_k$ are obtained by setting the theoretical probability equal to the empirical probability

$$\lambda_i / \sum_{j=1}^{k}\lambda_j = p_n(i) \quad i = 1, \ldots, k. \qquad (9)$$

It is easily seen that the λ's are not uniquely determined. This is true of the entire model; the scale parameters can be multiplied by a positive constant and the probabilities (1), (5)-(7) and the likelihood (8) are unchanged. Typically the parameters are chosen to satisfy the constraint $\sum_{i=1}^{k} \lambda_i = 1$. In this case the maximum likelihood estimates are $\hat{\lambda}_i = p_n(i)$. If the shape parameter is two, then the system of equations

$$\lambda_i^2 \left(k! \prod_{j \neq i} \lambda_j + \cdots + 2! \sum_{\substack{m=1 \\ m \neq i}}^{k} \lambda_m + 1\right) = p_n(i) \quad (10)$$
$$i = 1, \cdots, k$$

must be solved. An iterative algorithm is required to solve this system.

The Horse Race I roblem

The problem in which the marginal probability of finishing first is known for each object can be called the horse race problem since this is approximately the case at the racetrack. The argument supporting this statement is described in more detail later. The horse race problem is studied extensively by Ziemba and Hausch (1987). Fither the true probability that horse i wins or the empirical probability that horse i wins is assumed to be known for each of the k horses. Ziemba and Hausch then use the gamma model with shape parameter one to estimate the probability that each horse finishes second or third. They use these estimates to compute the expected return on place and show bets. A place bet on horse i is won if horse i finishes first or second. A show bet is won if the horse finishes third or higher. Here we compare the estimates provided by the model with r equal to one and the model with r equal to two.

The results of horse races at Bay Meadows racetrack in California during January and February 1987 were collected from the newspaper. For each race the track odds and the result of the race were recorded. The odds are determined by the amount of money bet on each horse to win the race. Approximately 20% of the wagered money is kept by the track in the form of the track take and breakage (rounding off the odds). The remaining money is split among those who bet on the winning horse in proportion to the size of their bet. Suppose that there are k horses and the odds for horse i are O_i:1. A \$2 bet on horse i to win the race returns \$ $2O_i + 2$ if horse i wins the race and nothing if horse i loses. Let $p_t(i)$ be the proportion of the total number of dollars bet on horses to win the race which is bet on horse i to win the race. Without the track take, $p_t(i)$ and the odds O_i would be related by $p_t(i) = (O_i + 1)^{-1}$. Since the reported odds are adjusted for the track take we take

$$p_t(i) = \frac{1/(O_i + 1)}{\sum_{j=1}^{k} 1/(O_j + 1)} \quad i = 1, \ldots, k.$$

The collection $p_t(i), i = 1, \ldots, k$ represents the probability estimate of all the bettors considered together. Ziemba and Hausch study the estimates $p_t(i)$ and find that they provide good estimates of the probability that horse i wins the race. They find that horses with high probability of winning win more often than predicted by the bettors and horses with low probability of winning win less often. People are attracted by the potentially large payoff in the latter case. Despite this inaccuracy, it is not possible to do much better than $p_t(i)$ as an estimate of the probability that horse i wins.

As in Harville (1973) and Ziemba and Hausch, we use $p_t(i)$ as the empirical probability although it is actually the public's estimate of the probability that i wins. Then $p_t(i)$ and the gamma model with shape parameter one are used to estimate the probability that horse i finishes second. From the previous section, the maximum likelihood estimates are

$$\hat{\lambda}_i = p_t(i), \qquad i = 1, \ldots, k.$$

Let $\hat{p}^{(1)}(\cdot\, i)$ be the estimated probability that i finishes second under this model. Then according to the Harville formulas (the gamma model with shape parameter one),

$$\hat{p}^{(1)}(\cdot\, i) = \sum_{j \neq i} \hat{\lambda}_j \frac{\hat{\lambda}_i}{1 - \hat{\lambda}_j} = \sum_{j \neq i} p_t(j) \frac{p_t(i)}{1 - p_t(j)}.$$

As an alternative we fit the gamma model with shape parameter two to the same data. We consider $p_t(i)$ fixed and use an iterative procedure to solve the likelihood equations (10) for the maximum likelihood estimates. The estimated probability that i finishes second according to this gamma model is computed by summing the estimated probability of all permutations in which i finishes second. Let the estimates from this gamma model be $\hat{p}^{(2)}(\cdot\, i)$. The estimates for both models computed for a sample race are:

Sample Horse Race Data

2^{nd} Race, January 9,1987

Horse	O_i	$p_t(i)$	$\hat{p}^{(1)}(\cdot\, i)$	$\hat{p}^{(2)}(\cdot\, i)$
1	8.9	0.084	0.107	0.113
2	19.8	0.040	0.053	0.059
3	1.1	0.395	0.280	0.266
4	3.3	0.193	0.217	0.214
5	5.2	0.134	0.162	0.165
6	4.4	0.154	0.182	0.183

Those horses with large probabilities of winning have a lower probability of finishing second when r equals two.

Due to the complicated calculation required to compute the estimates when the shape parameter is two, we restrict attention to 47 races in which there were 6 horses. For each of the 47 races, estimates of the probability that each horse finishes second are available under two different models. The true probability that each horse finishes second is unknown, only the identity of the actual second place horse is available. The number of times that a horse with a given estimated probability of finishing second actually finished second is compared to the expected number below. There are a total of 282 horses in the 47 races. Horses with similar estimated probabilities of finishing second are grouped together. The expected number of second place finishes for a particular group is computed as the sum of the estimated probabilities for the horses in that group.

Expected and Observed Second Place Finishes

$\hat{p}$	$r = 1$ Observed	$r = 1$ Expected	$r = 2$ Observed	$r = 2$ Expected
0.00-0.10	6	4.15	4	3.75
0.10-0.15	7	7.69	9	7.94
0.15-0.20	13	8.56	13	9.97
0.20-0.25	12	12.54	13	14.10
>0.25	9	14.06	8	11.23

Note that the observed second place horse is put into a group based on the estimated probability of its finishing second. The same second place horse may be in different groups under the two gamma models. In the first pair of columns we observe results similar to those of Harville (1973). Horses which have high predicted probability of finishing second finish second less often than predicted. Horses with low estimated probability of finishing second do better than expected. By taking r equal to two, the fit in these two areas is improved. It seems that higher values of r might lead to better fits but this has not been verified.

Summary

Comparisons of gamma random variables with a given shape parameter provide a family of distributions for permutations of k objects. When these models are restricted to the paired comparison experiment they provide alternative derivations of the currently used models. These models are then naturally generalized to the problem in which k objects are compared at a time. There are indications that large sample sizes are needed to distinguish between the models which correspond to different shape parameters. However there are also indications that models with shape parameter other than one can be successfully applied in practical problems. In order to make the models more widely applicable improved methods for computing the distributions are required. It would be particularly useful to find approximations which can be easily computed.

References

Bradley, R. A. (1953). Some Statistical Methods in Taste Testing and Quality Evaluation. *Biometrics* **9**, 22-38.

Bradley, R. A. (1954). Incomplete Block Rank Analysis: On the Appropriateness of the Model for a Method of Paired Comparisons. *Biometrics* **10**, 375-390.

Bradley, R. A. (1955). Rank Analysis of Incomplete Block Designs. III. Some Large-Sample Results on Estimation and Power for a Method of Paired Comparisons. *Biometrika* **42**, 450-470.

Bradley, R. A. (1965). Another Interpretation of a Model for Paired Comparisons. *Psychometrika* **30**, 315-318.

Bradley, R. A. and Terry, M. E. (1952). Rank Analysis of Incomplete Block Designs. I. The Method of Paired Comparisons. *Biometrika* **39**, 324-345.

Burke, C. J. and Zinnes, J. L. (1965). A Paired Comparison of Paired Comparisons. *J. Math. Psych.* **2**, 53-76.

Daniels, H. E. (1950). Rank Correlation and Population Models. *J. R. Statist. Soc. B* **12**, 171-181.

David, H. A. (1963). *The Method of Paired Comparisons*, Griffin, London.

Davidson, R. R. and Farquhar, P. H. (1976). A Bibliography on the Method of Paired Comparisons. *Biometrics* **32**, 241-252.

Davidson, R. R. and Solomon, D. L. (1973). A Bayesian approach to Paired Comparison Experimentation. *Biometrika* **60**, 477-487.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Statist. Soc. B* **39**, 1-38 (with discussion).

Ford, L. R. Jr. (1957). Solution of a Ranking Problem from Binary Comparisons. *American Math. Monthly* **64**, 28-33.

Harville, D. A. (1973). Assigning Probabilities to the Outcomes of Multi-Entry Competitions. *J. Amer. Stat. Assoc.* **68**, 312-316.

Henery, R. J. (1981). Permutation Probabilities as Models for Horse Races. *J. R. Statist. Soc. B* **43**, 86-91.

Henery, R. J. (1983). Permutation Probabilities for Gamma Random Variables. *J. Appl. Prob.* **20**, 822-834.

Jackson, J. E. and Fleckenstein, M. (1957). An Evaluation of Some Statistical Techniques Used in the Analysis of Paired Comparisons. *Biometrics* **13**, 51-64.

Latta, R. B. (1979). Composition Rules for Probabilities from Paired Comparisons. *Ann. Statist.* **7**, 349-371.

Lehmann, E. L. (1983). *Theory of Point Estimation*, John Wiley, New York.

Leonard, T. (1977). An Alternative Bayesian approach to the Bradley-Terry Model for Paired Comparisons. *Biometrics* **33**, 121-132.

Mallows, C. L. (1957). Non-Null Ranking Models. I. *Biometrika* **44**, 114-130.

Mosteller, F. (1951). Remarks on the Methods of Paired Comparisons: I. The Least Squares Solution Assuming Equal Standard Deviations and Equal Correlations. II. The Effect of an Aberrant Standard Deviation When Equal Standard Deviations and Equal Correlations are Assumed. III. A Test of Significance for Paired Comparisons when Equal Standard Deviations and Equal Correlations are Assumed. *Psychometrika* **16**, 3-9, 203-206, 207-218.

Stern, H. S. (1987). *Gamma Processes, Paired Comparisons, and Ranking*. PhD Thesis, Department of Statistics, Stanford University.

Thurstone, L. L. (1927). A Law of Comparative Judgment. *Psychol. Rev.* **34**, 273-286.

Zermelo, E. (1929). Die Berechnung Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Math. Zeit.* **29**, 436-460.

Ziemba, W. T. and Hausch, D. B. (1987). *Dr. Z's Beat the Racetrack*. William Morrow and Company, New York.

A MODULAR NONPARAMETRIC APPROACH TO MODEL SELECTION

Michael E: Tarter and Michael D. Lock

Department of Biomedical and Environmental Health Sciences,
University of California, Berkeley

A two stage approach is introduced which allows a researcher to choose from among ten alternative families of models for conditional "slices" and individual component "sections," of a mixture of marginal or conditional densities. The general concept of logmodel is introduced and it is considered that the pair of models, normal-lognormal is only one of at least five classes of general logmodel systems. A functional, referred to as R(x), is introduced which allows one to determine model-class membership without the need to conduct multiple trials with arbitrarily selected pretransformation location parameters, e.g., the constant C of the transform Log(Y - C). The R(x) functional is applied to the problem of parameter estimation for a system of models which is the dual of the Johnson family of models. Conditions for the existence of the above type of functional are derived and an example of a model is given for which it is shown that no functional exists which has the properties of R(x) constructed from logmodel systems.

KEY WORDS

Lognormal; Logmodel; Loglogmodel; Conditional estimation; Mixture decomposition; Transformations; Model-free methods.

0. Introduction: It is common in the early stages of a statistical investigation to construct a histogram or scatter diagram. This process is usually performed for the purpose of checking the assumptions upon which later stages of the data analysis process will be based. Note the important fact, that when the statistician plots a histogram, it is usually the nature of a marginal and not a conditional density that is being checked. However, it is more often a conditional density, component of a mixture of marginal densities or even a mixture of conditional densities that is the most appropriate target of the model selection process.

Consider, for example, the simple case of linear regression where the observations are divided into two sets of replicates, i.e.,

$$E(Y_{ij}) = \alpha + \beta X_i,\ i=1,2;\ j=1,..,n_i,$$

where X_1 is not equal to X_2 and β, n_1 and n_2 are nonzero. Even in the commonly assumed situation where for a given value of X_1 or X_2, the variate Y_{ij} is normally distributed, the marginal of the population of all Y_{ij} variates will be a mixture of normals. Hence, in order to check assumptions about Y_{ij}, the researcher faces the task of characterizing a multicomponent distribution. In the above scenario the researcher may be able to circumvent the difficulty of dealing with a mixture of distributions by examining the conditional distributions of Y given X_1 and Y given X_2, for example by constructing two histograms. However, in many realistic data analysis situations one will not initially be aware of the existence of a variate whose values, like $X_1 = x_1$ and $X_2 = x_2$ index distributional identity. Even in the situation where the existence of such a variate is known, unless large numbers of replicates are available at particular values of this variate, the investigator must rely on the analysis of regression residuals to check assumptions about the underlying distribution. One trouble with this approach is, that in order to analyze the distribution of residuals about a regression line, an investigator must rely on some simple and usually linear assumption about the functional $E(Y|x)$. Useful though it is in some situations, residual analysis often leaves one uncertain as to whether it is the model for $E(Y|x)$ itself, or the model for the distribution of the residuals from $E(Y|x)$ or perhaps both models that are suspect.

In addition to this difficulty, another problem is sometimes encountered in conventional regression analysis. As illustrated in Tarter, Pollisar and Freeman (1983) the relationship between

two variates may not only be nonlinear, but a single-valued function which adequately describes this relationship may not even exist. In view of the substantial interest in mixture decomposition and cluster analysis techniques, it seems understandable that any study based solely on an analysis of the residuals from a single-valued locus of $E(Y|x)$ evaluations may be overly simplistic.

The methodology described in this paper seeks to overcome the difficulties inherent in the use of residual analysis to characterize the underlying distribution, as well as to investigate the possible existence of unanticipated independent variates. A systematic process allowing the researcher to select an appropriate model or transformation for his or her data is proposed.

The methods detailed in sections 3 and 4 are all two step processes. The first step always consists of estimating the entire joint distribution of the observations in as general a manner as is possible, given the amount of available data. Since all properties, curves and functionals of statistical interest can be defined in terms of the joint distribution of all variates, once the underlying distribution function is estimated, one can proceed to the second step of obtaining what can be referred to as "secondary" estimators.

The problem targeted by this paper is: How can secondary estimators be used to select from among a wide spectrum of distributional models? In particular, one would like the search for an answer to this question to proceed in a systematic fashion which takes advantage of the interrelationships between certain general classes of statistical models.

1.Models and Logmodels: One basic characteristic differentiates the proposed model selection process from previously considered procedures such as those treated extensively by Johnson (1949). As detailed in section 4, the logarithmic data transformation will be a common element. Thus, the new methodology can be interpreted as being the dual of the Johnson approach in the sense that one commonly used data transformation will be associated with a spectrum of models. (The Johnson approach can be characterized as the consideration of a variety of transformations to yield data which conforms to a single model, the normal.) As suggested by Thompson (1988), the Johnson "family" was proposed at a time when the normal model held a much more prominent position in the pantheon of underlying distributions than it does today. Many of the applied scientists with whom the authors work make extensive use of the logarithmic transformation. Thus, while we feel that neither the logarithm or the normal distribution is in any sense sacred, we do feel that there is as much justification for the selection of a single transformation as a common element as there is for selecting a single model.

As an example of this approach consider, as will be detailed in section 4, that one can interpret the relationship between the standard uniform or rectangular density model and the exponential model, to be analogous to the relationship between the lognormal and the normal model. The negative logarithm of a standard uniform variate will yield a standard exponential variate in exactly the same way that the logarithm of a lognormally distributed variate will yield a normal variate. Thus, the details of a process by which one can conduct a search for the most appropriate model will be given in section 4. This process takes advantage of the fact that not only the rectangular distribution, but also the exponential distribution, can be treated as a special case of the general power distribution family of models.

There are several other commonly encountered pairs of models which have the above type of relationship. The logarithm of a Weibull variate minus a constant is known to be an extreme value variate (Johnson and Kotz, 1970, page 272). There has also been some consideration of what could be called a loglogistic model and, as will be shown below, there is no reason why one cannot conceptualize a logcauchy model.

The lognormal is obviously by far the most commonly assumed of the logmodels. It is also very often used inappropriately due to the need in many situations, to estimate an appropriate constant for subtraction from the pretransformed variate as a preliminary to calls to the log function. It will be demonstrated below that this constant plays an important role in other logmodels and thus, for the purpose of ease of reference, this constant will be called a "pretransformation location parameter", symbolized by μ_1. For completeness, once the exact value of the pretransformation location parameter has been subtracted from the lognormally distributed variate, the natural log of the difference will be said to be distributed with scale parameter σ and "posttransformation location parameter" μ_2.

A common graphical procedure which is often used to check on the validity of

the lognormality assumption, is the lognormal plot. Even after the advent of sophisticated personal computer graphical packages, one still might find it useful to hand-plot an estimated cdf on graph paper whose abscissa is graduated by a log scale and whose ordinate is graduated on a standard normal cumulative scale. An example of such paper is given in Dixon and Massey (1983) page 488.

One could select the particular functional associated with the lognormal plot as what was previously referred to as a "secondary estimator." However, when implemented in the usual way, there is an important weakness to this approach which, as will be shown below, is not associated with a second, and closely related functional.

Consider that the lognormal plot utilizes the human visual system's sensitivity to the straightness or lack of straightness of a line and that, as is well known, in two dimensions, a line is characterized by two numbers, for example a slope α and a Y-intercept β. The lognormal model is characterized by <u>three</u> numbers: a scale parameter, and two location parameters, μ_1 and μ_2. For the following reason the posttransformation location parameter μ_2 is much easier to estimate than the pretransformation location parameter μ_1:

In the case of lognormal data, once μ_1 is determined, one can take the logarithm of the underlying variate minus this known value and be certain that the resulting variate will have a normal distribution (or in the case of the four other distribution systems considered below, an exponential, extreme value, logistic or Cauchy distribution). In other words, once one's data have been transformed to normality, the estimation of the posttransformation location parameter μ_2 has been reduced to the extremely well studied problem of estimating the mean of a normal variate. Hence, one finds that a kind of Catch-22 applies to lognormal data analysis. The normal probability plot can be used to ascertain the value of the easy-to-estimate parameter μ_2, but requires the clumsy expedient of repeated trials with a variety of choices for the value of μ_1 in order to choose between alternative values of the μ_1 parameter. Clearly, it would be preferable for a graphical procedure to estimate the values of μ_1 and σ, rather than the values of μ_2 and σ, and leave the problem of μ_2 estimation to the solution of an easily solved, rather than, as is the case of μ_1 estimation, the solution of a difficult problem (Cohen 1951, Aitchison and Brown 1957).

If as an alternative to the functional which underlies the lognormal plot, one constructs the functional R(x) by following the following two steps: Step 1; substitute the unknown cumulative $y = F(x)$ into y, (to detect membership within the power-exponential family), into minus ylog(y) (to detect membership within the Weibull-extreme value family), into y(1-y) (to detect membership within the loglogistic family), into $(1+[\tan\pi(x-1/2)]^2)^{-1}/\pi$ (to detect membership within the logcauchy family); and Step 2; divide the above by the unknown density f(x) where $f(x) = F'(x)$ one finds the following:

1) Whenever the true distribution is exponential, (or for other special cases of R(x), extreme-value, logistic or Cauchy) the functional will be identical to a horizontal line of height equal to the value assumed by the scale parameter of the model.

2) Whenever the true distribution is power, (or Weibull, loglogistic or logcauchy) the functional will be a diagonal line whose slope and Y-intercept are determined by the scale parameter and pretransformation location parameter μ_1.

3) In the above cases, the R(x) functional will be unrelated to the value of the posttransformation location parameter μ_2.

4) The above properties apply to the normal and lognormal models for the special case of the R(x) functional described in section 3 of this paper, which, unlike the other four examples described in this paper, cannot be expressed in closed form in terms of elementary functions but instead must be expressed in terms of the normal inverse cumulative.

2. <u>The Logtranformation as C increases</u>: We will now show that in one important sense, the role played by the pretransformation location parameter $\mu_1 = C$ is in actuality the opposite of what one might suspect it to be. It is a large value for C rather than a small value which yields a log transform which minimally affects the pretransformed variate. This observation can be considered to be a corollary to the following theorem:

A lognormal cumulative with pretransformation location parameter -C, posttransformation location parameter [log C + μ/C] and scale parameter σ/C, approaches a normal cumulative with location parameter μ and scale parameter σ as C approaches <u>infinity</u>.

Proof: Consider that a lognormal cumulative F(x), as defined above, can be expressed as $\Phi(v(x))$ where v(x) is identical to $([\log(x+C) - \log C]/C^{-1} - \mu)/\sigma$ and Φ represents the standard normal cumulative. By applying l'Hospital's

Rule twice, one finds that the limit of $[\log(x+C) - \log C]/C^{-1}$ as C approaches infinity equals x. Now consider the following series expansion for $\Phi(x)$ given in Kendall and Stuart (1958) p 136: $\Phi(x) = (1/2)+ (2\pi)^{-1/2}(x-[x^3/6][1-(3/10)(x^2/2!)+...])$ Each term of the bracketed series is less than or equal in value to a consecutive term of the power series expansion of cos(x). By Abel's Theorem the power series expansion of cos(x) converges uniformly for any value of x and thus, by the comparison test, the above power series expansion of $\Phi(x)$ is everywhere uniformly convergent.
Let $\Phi_n(x)$ represent the n-term partial sum of the above power series. Consider the double limit Lim Lim $\Phi_n((\log(x+C)-[\log C+\mu/C]\}/(\sigma/C))$, expression (1), where the inner limit is taken as C approaches ∞ and the outer limit is taken as n approaches ∞.
Each individual $\Phi_n(x)$ is a finite sum of powers of the curly bracketed expression and thus the inner limit equals $\Phi_n((x-\mu)/\sigma)$ since the limit of $C[\log(x+C) - \log C]$, as C approaches ∞, was shown above to equal x. From Theorem 7.11 of Rudin(1953), since Φ_n approaches Φ uniformly, the order of the two limits of expression (1) can be reversed which proves the theorem.

Since unlike the normal cumulative Φ, the other four logmodels considered in this paper have cumulative distribution functions which are closed forms of elementary functions, one does not need the elaborate argument presented above to prove that the above feature of the limit of the log transformation as C approaches infinity applies to these models.

3. Classes of Logmodels and the R(x) functional: A. C. Cohen's (1951) pioneering paper begins with the sentence: "The logarithmic normal distribution provides a useful theoretical model for studying a number of biological populations, certain economic populations involving income distributions, and others in which the standard deviation of individual observations is approximately proportional to the magnitude of the observations." The last part of this sentence is puzzling since it is hard to see how an individual observation can have a standard deviation. However, it will be shown below that: There is a property of the lognormal model which is both linear and connected with the standard deviation. There is exact, as opposed to approximate proportionality involved; and, in the opinion of the authors most importantly, this property is shared with at least four other important classes of distributional models.
In the remainder of this paper the following notation and assumptions will be utilized:
1) The symbols ϕ, Φ and Φ^{-1} will represent the standard normal, i.e. Gaussian, density, cumulative and inverse cumulative respectively.
2) The symbols g, G and G^{-1} will represent respectively the standard forms of the density, cumulative or inverse cumulative of any one of the following five distributions: Positive exponential, extreme value, logistic, Cauchy or normal, defined over an interval (a,b) where $G^{-1}G(x)$ is identically equal to x for any x within (a,b).
3) The symbols f, F and F^{-1} will represent respectively the density, cumulative and inverse cumulative such that for some G defined above, F(x) is identically equal to $G([\log(x-\mu_1)-\mu_2]/\sigma)$. Tarter and Kowalski (1972) considered the case where G is identical to Φ. It was shown that the functional R(x) defined as $\phi\Phi^{-1}$ $F(x)/f(x)$ has the following three properties:
PROPERTY 1: $R(x) = \sigma$ for all finite x if and only if F(x) is a normal cumulative with standard deviation σ.
PROPERTY 2: $R(x) = \sigma(x-\alpha)$ for any $x > \alpha$ and zero elsewhere if and only if F(x) is a three parameter lognormal distribution function as defined above for G identically equal to Φ.
PROPERTY 3: $R(x) = \sigma(x-\alpha)^2$ for all finite x if and only if F(x) is a three-parameter reciprocal normal distribution function, i.e., if and only if $F(x) = [((\alpha-x)^{-1} - \mu)/\sigma]$.
PROPERTY 4: $R(x) = (x-\mu_1)(\mu_2-x)/(\mu_1-\mu_2)$ for $\mu_1<x<\mu_2$ and zero elsewhere if and only if F(x) is a distribution with associated random variate X which can be transformed to a normal variate Z by the transformation $Z = \mathrm{Log}((X-\mu_1)/(\mu_2-X))$.
The remainder of this paper will consider choices of the function gG^{-1} within the definition of R(x), other than the function $\phi\Phi^{-1}$. It will be shown that properties one and two have several useful extensions to a variety of statistical models. However, before turning to applications of gG^{-1} alternatives, it seems useful to reconsider the Cohen statement that the "standard deviation of individual observations is approximately proportional to the magnitude of the observations."

Notice that one can treat PROPERTY 1 as the limiting case of PROPERTY 2. This observation implies in turn that the normal density can be treated as a special case of the lognormal density. (Note that a linear R(x) with an extremely small but nonzero slope

characterizes a lognormal, while a perfectly horizontal line characterizes a normal model).

There is a concrete way to view the relationship between F-type and G-type models where $F(x)=G([\log(x-\mu_1)-\mu_2]/\sigma)$ and G represents the cdf of either an exponential, extreme value, logistic, Cauchy or normal variate. Consider a technique for simulating data from a specified distribution referred to in Tocher (1963) pages 22-24 as the "Method of Mixtures." The method is based in part on partitioning the support region of the distribution from which data are to be simulated into class intervals, in a manner similar to the procedure by which conventional histograms are constructed.

Let the symbol x_a represent an arbitrary left end-point of a class interval and $x_a+\epsilon$ represent the right end-point of this same interval, where of course, $\epsilon > 0$. Furthermore, define $A = G([\log(x_a-\mu_1)-\mu_2]/\sigma)$ and $B = G([\log(x_a+\epsilon-\mu_1)-\mu_2]/\sigma)$ to be the areas under the density from which data are to be simulated, to the left of the left and right class interval endpoints respectively. Now suppose that as one would do by the method of mixtures, one wishes to simulate data from one of the above logmodels by using data generated from the associated model. To assure that the probability mass over the above class interval equaled B-A, one could solve the system of two equations, $\sigma_a G^{-1}(A) + \mu a = x_a$ and $\sigma_a G^{-1}(B) + \mu a = x_a + \epsilon$ for the constants μ_a and σ_a and find that $\sigma_a = \epsilon/[(\log(x_a+\epsilon-\mu_1)-\log(x_a-\mu_1))/\sigma]$. If one applies l'Hospital's rule one finds that as the length of the interval, ϵ, approaches zero, the scale parameter σ_a approaches $(x_a-\mu_1)\sigma$, i.e., one can use the method of mixtures to simulate data from any logmodel by using data generated from the associated parent model and linearly varying the scale parameter of the parent model data.

An alternative view which can be used to visualize the above relationship is as follows: For a small positive value of ϵ, consider $R(x) = gG^{-1}F(x)/f(x)$ for x within the interval $[x_a,x_a+\epsilon]$. Since as defined above, F is the cdf of either a power, Weibull, loglogistic, logcauchy or lognormal variate with scale parameter σ, R(x) will be a line with slope equal to . Consider the line segment that connects the point $(x_a,R(x_a))$ to the point $(x_a+\epsilon,R(x_a+\epsilon))$. Suppose that this line segment is rotated about the point $(x_a+\epsilon/2,R(x_a+\epsilon/2))$ in order to form a horizontal line segment whose height is identically equal to $R(x_a+\epsilon/2)$. By the uniqueness of the general solution of the first order differential equation which defines R(x), the only analytic cdf which can determine such a horizontal R(x) within the small interval, must be identical to $F(x) = G([x-\mu_3]/\sigma_a)$ for some value of μ_3 and for σ_a identically equal to $(x_a+\epsilon/2-\mu_1)$ times σ. In other words, the scale parameter σ_a is a linear function of x_a.

The above explains the previously referred to characteristic of the lognormal model suggested by Cohen. and shows that this characteristic is shared by all logmodels, each of which can be thought of as a composition of "parent" model evaluations where the scale parameter of the parent model increases linearly. In essence, the R(x) functional, is this linear relationship. The term "parent" model corresponds to some G defined above, while the term "composition" refers to the limit, as ϵ approaches infinity, of some function which in any interval $[x_a,x_a+\epsilon]$ is identical to $G([x-\mu_3]/\sigma_a)$.

To clarify Cohen's statement, it is not the data points themselves which have an increasing scale parameter, but the pieces of the above-defined density mixture. Furthermore, all five of the logmodels considered in this paper, and not simply the lognormal, have this relationship. One can of course generalize Cohen's statement for the case of any cumulative $G([v(x_a-\mu_1)-\mu_2]/\sigma)$ where v is monotonic function with a first derivative v' for which $\sigma_a = \sigma[v'(x_a-\mu_1)]^{-1}$. It is the fact that the reciprocal of the derivative of the function $v(x) = \log(x)$ equals x, that underlies the convenience of the R(x) graphical approach.

4. Examples of logmodels: In order to demonstrate the generality of the modular estimation of R(x), it seems appropriate to detail the application of this functional to the goodness-of-fit of a sequence of model families more general than the five models described in section 1. Suppose ϕ, Φ, G, g, F, and f are all defined as in the previous section except that

$$F(x) = H[(\text{Log}(\text{Log}(x-\mu_1)-\mu_2)-\mu_3)/\sigma]$$
$$= H[\text{Log}(\text{Log}(x-\mu_1)-\mu_2)^{1/\sigma}-\mu_3/\sigma]$$

where $G(x) = H[\text{Log}x^{1/\sigma}] = H((1/\sigma)\text{Log}(x))$ and H, rather than G, is a standard exponential, extreme value, logistic Cauchy or normal cdf. For these "loglogmodels," the gG^{-1} component of the R(x) functional is identical to $hH^{-1}(y)\exp[H^{-1}(y)]$. Hence, not only can one apply the R(x) functional to the problem of graphically selecting between five commonly used logmodels f, but one can use an approach similar to the conventional application of the lognormal probability plot to estimate σ and the three additional parameters μ_j,

j = 1,2,3 of any loglogmodel. Instead of selecting a trial value for the pretransformation location parameter μ_1 and checking the straightness of the resulting lognormal probability plot, one can: 1) select a trial value for the scale parameter σ;
2) check by the straightness of the R(x) functional on the accuracy of this choice for σ;
3) when a straight R(x) is obtained, one can estimate the parameters μ_1 and μ_2 in terms of the estimated slope and y-intercept to R(x);
4) in the univariate case, once μ_1 is estimated, one can transform each data point X_i to $Log[X_i - \mu_1^*]$ where μ_1^* represents the graphically obtained estimator of μ_1 and finally;
5) now that the problem of estimating the parameters of a loglogmodel has been reduced to the problem of estimating the parameters of a logmodel, one can use the R(x) functional again to check, by the straightness of R(x), on the validity of the choice of the underlying h as one of the five model families considered above.

It is appropriate to point out that by extending the derivation detailed in section 3, one can show that each logmodel is a special limiting case of a loglogmodel. Consider the Pearson family and most other methods used to generalize the normal model. In the case of the Pearson family, it is the coefficients of terms in a quadratic denominator whose common zero value reduces the general model to the special normal case (Elderton, 1953, p.49). In both the logmodel and loglogmodel systems, it is the value infinity which "reduces" the general model to its special case. However, unlike the Pearson family and any other generalized model family with which we are now aware, only the loglogmodel system has both the normal and the lognormal as special cases.

It is also of some interest to compare the approach to model generality considered in this section to the Johnson Family of Distributions (Johnson 1949, Tapia and Thompson, 1978, p 30-33). Consider that there are two basic assumptions involved in use of the lognormal model. It is assumed that:(1) a logarithmic transformation will, (2) transform one's data to normality. In essence, the Johnson Family approach generalizes assumption (1) while the methodology considered here generalizes assumption (2). In the next section of this paper the question of the generality of the R(x) function itself will be considered and it will be asserted that the connection between this type of graphical method and the logarithmic transformation is not necessarily shared by alternative distribution systems.

5. Conditions for the Existence of an R(x)-type Graphical Method: In order to consider the general properties of the previously described modular methods, it is useful to represent the functional R by $R[f(x;\theta_1,\theta_2),F(x;\theta_1,\theta_2)]$ where f and F represent the hypothesized pdf and cdf of the underlying random variate which are specified up to the values of the two unknown parameters θ_1 and θ_2. The following question then arises: For what classes of statistical models, will there exist a differentiable function R with the following two properties:
1; For a fixed value of θ_2, R will be identically equal to a constant for all values of the random variate X = x.
2; For a fixed value of θ_2, the value of R will not change, i.e. R will assume a constant value, for any value assumed by the parameter θ_1. These are two of the properties which make the R(x) functional particularly useful. For example, in the case of the normal model $\{\phi[(x-\mu)/\sigma]\}/\sigma$, $\theta_1 = \mu$ and $\theta_2 = \sigma$, the value assumed by R is equal to σ for any value of x, i.e., Property 1, and R is functionally unrelated to μ , i.e., Property 2.

If one restricts R and F to the class of differentiable functions, it is easy to obtain necessary conditions for R to have Properties 1 and 2. Let u and v represent the two arguments of the function R. By the chain rule, Properties 1 and 2 imply that
$\{\delta R/\delta u\}\{\delta f(x;\theta_1,\theta_2)/\delta x\}+\{\delta R/\delta v\}$
$f(x;\theta_1,\theta_2)=0$
$\{\delta R/\delta u\}\{\delta f(x:\theta_1,\theta_2)/\delta\theta_1\}$ +

$\{\delta R/\delta v\}\{\delta F(x;\theta_1,\theta_2)/\delta\theta_1\} = 0$
The truncated exponential model provides an example of a parametric family which does not satisfy the above necessary conditions. Specifically, suppose $f(x) = (b/\sigma)\exp(-x/\sigma)I_{[0,B]}(x)$, where $I_{[0,B]}$ represents the indicator function of the closed interval [0,B], and b is chosen to assure that the definite integral of f over [0,B] equals one. For this choice of f, the above necessary conditions imply that $\{\delta R/\delta u\}\{-b/\sigma^2\} + \{\delta R/\delta v\}\{b/\sigma\} = 0$ and
$\{\delta R/\delta u\}\{(1/\sigma)\exp(-x/\sigma)+$
$\{\delta R/\delta v\}\{1-\exp(-x/\sigma)\}=0$,
which in turn implies that $-\exp(-x/\sigma)/[1-\exp(-x/\sigma)]$ identically equals one. Hence, there cannot exist a differentiable function R which has properties 1 and 2 for the special case of the truncated exponential model. On the other hand, for any functions g, G and G^{-1} defined in section 3 and h, H and H^{-1} where $h(x) = g([x-\mu]/\sigma)/\sigma$, $H(x) = G([x-\mu]/\sigma)$ and $H^{-1}(y) = \mu + \sigma G^{-1}(y)$, the necessary conditions imply that $\{\delta R/\delta u\}/\{\delta R/\delta v\}$ equals $-h(x)/h'(x)$, which is satisfied by $R(u,v) = gG^{-1}(v)/u$.

6. A Description of the Modular Model Selection, Slicing and Sectioning Program: The principal goal set during the construction of this program was that the model selection process was to be based on either an estimated slice, i.e. conditional; section, i.e., separated distributional subcomponent or; (and in the authors' opinion, the least applicable case) an estimated marginal. Thus, the program differs radically from previous goodness-of-fit subroutines, such as that included within STATGRAF (1988) which can only be used to check on model appropriateness for univariate data.

The main menu of the program allows a user to select a file of univariate or bivariate data and estimate a marginal of either of two variates or alternatively estimate a conditional slice of one variate given any selected value of the second variate. The first call to the conditional slice subroutine initiates the computation of those bivariate sample Fourier coefficients (trigonometric moments) found to be appropriate for the estimated distribution and available sample size. (Tarter and Kronmal 1970, describes the theory which underlies this procedure.) For samples larger than one thousand from multicomponent mixtures, the execution of this one step can take as long as three minutes on an IBM AT compatible PC with a 6 MHz clock. However, since for any given set of data, all subsequent slicing, sectioning and model selection procedures are based on this same set of estimated Fourier coefficients, all further calculations can be performed with only from two to fifteen second response time delays on the 6MHz AT-type PC.

One of the main menu options is labeled CONTRAST. This option allows a user to enter a value of a constant which separates individual estimated distributional components. (Details of this procedure, but not the model selection process, are described in Tarter 1979, section 3.) If the CONTRAST option is selected immediately after a conditional slice has been estimated, a user can section a conditional in exactly the same way that he or she can section a marginal, if the CONTRAST option is executed immediately after a marginal is estimated.

The two options listed in the program's main menu which are most germane to this paper are labeled R(X) and FIT MODEL. Each of the five model-logmodel families considered above is keyed to a particular color. By using a combination of R(X) and MODEL SELECTION options, a user can graph any combination of estimated R(x) functionals or fitted models or logmodels. The choice of color is used to identify the model upon which an estimated R(x) and fitted density are based. A dotted line is used to graph an estimated logmodel while a dashed line of the same color is used to identify the associated model.

One major practical problem was encountered when the primary bivariate estimation subroutines were combined with the sectioning, slicing and R(X) model identification subroutines. The standard forms used to represent each of the five model systems considered in section 4 in no way took into account the problem of comparing the fit of these models. One of the authors of this paper had encountered this difficulty twice in the past. In Kronmal and Tarter (1968) it was found that in order to compare the Mean Integrated Squared Error characteristics of nonparametric estimates obtained from normal and Cauchy data, one needed to introduce a variant of the Cauchy "standard model" that was comparable to the standard normal, Φ, in the sense that the two standard-form models had the same first, second and third quartiles. In Tarter (1968), the optimal scale parameter coefficient of the standard logit, $\log(y/(1 - y))$ was found which provides the best fit of this model to the inverse standard normal cumulative based on the integrated squared error metric.

It might also be pointed out that the constant two, chosen as a divisor of $-x^2$ within the "standard" normal exponent $-x^2/2$, serves the sole purpose of assuring that the scale parameter σ of the nonstandard normal is identical to the property of the distribution usually referred to as the "standard" deviation. In fields such as optics, where the normal is used without need for the convenience of scale parameter and standard deviation identity, the standard distribution is defined without the constant two.

Since the property, standard deviation, does not exist for the Cauchy model, the standard form of this distribution is usually chosen for reasons of simplicity and unlike the normal, no attempt is made to identify the scale parameter with a property of the distribution. This means that there is no particular reason to expect that the standard form of the Cauchy will be in any way comparable to the standard form of the normal.

The logistic inverse cumulative, i.e., $\log(y/[1 - y])$, is so simple a function, that no attempt is usually made to see that the scale parameter of this model is identical to any distributional property or comparable to the standard normal inverse cumulative Φ^{-1}. Only the

exponential model seems to share with the normal the distinction that its standard form allows a parameter of the nonstandard form to be associated with an easily interpreted property. Specifically, if $f(t)=\exp(-\tau t)I_{[0,\infty]}(t)$, then $\tau=1/E(t)$, Gross and Clark (1975), p.52.

One can use the fact that families of logmodels have the above-mentioned arbitrariness to one's advantage, by comparing two alternative R(x) functionals which are graphed so that they assume approximately the same value at the estimated mode of one of the densities. In the case where a model is correctly selected, Property 2 of section 3 implies that the true R(x) functional will be a line with slope identically equal to the scale parameter of the model. It should also be mentioned that the above dependence of a graphical method on the definition of the "standard model" is shared by conventional graphical methods which rely on a plot of estimated cumulative probability on a y-axis which has been transformed to "standard" normal scale, against a transformed scale, e.g., log or square root, on the x-axis, as illustrated in Dixon and Massey (1983) page 488. If one were to prepare Cauchy or logistic probability paper to compare normal to Cauchy or logistic fit, one would find that the slope of the plot obtained under the null hypothesis of true fit, would be dependent upon the definition of scale parameter of the nonstandard model, i.e., F.

The R(x) functional has a slope equal to zero in the null hypothesis case for any of the five parent models considered in this paper. Hence, besides being far less dependent on choice of standard model than is the conventional lognormal-type plot, it also tends to circumvent a subtle problem associated with the Kolmogoroff-Smirnoff, K-S, or Chi-square goodness of fit procedures included in such packages as STATGRAF (1986). To use these two procedures, the parameters of the "fitted" models must be estimated before comparisons can be made. In essence, this restriction confounds the problem of estimator efficiency with the problem of model specification. (Tapia and Thompson 1978 section 1.4 contains an excellent description of this form of confounding). The same estimators of the pdf f and cdf F are used to estimate all R(x) curves and thus, the methods described in this paper tend to circumvent the problem of specification, parameter-estimation confounding.

If the estimator of f or F is poor at a distance plus or minus C from the estimated mode of f, one can simply shorten the region over which R(x) is estimated by changing K to some smaller value, e.g., 2K/3. Due to the choice of the mean integrated squared error metric, MISE, which underlies most kernel and series density estimation procedures, estimators can be said to be "center-weighted," in much the same way that the automatic exposure meters built into many current 35mm cameras are designed to provide the most accurate estimates of light conditions near the image center. Thus, one can be fairly sure that as the constant K referred to above is reduced, one's estimator accuracy will improve. Of course there are limits to the effectiveness of this procedure since, as is true in photography, peripheral information may be important even if it is slightly out of focus.

In the context of alternatives to the R(x) approach, one could choose to modify the Chi-squared goodness of fit test, and only compare observed to fitted model frequencies within a subinterval of the model's support. However, when one chooses to do this, one is faced with the dilemma of whether or not one should base the estimation of the parameters of the fitted model upon a data set which is censored to solely include values which lie within the restricted subinterval. In the univariate case, one could attempt to use BLU procedures to obtain estimates, but in the case where it is a truncated conditional, i.e., slice or single section of a multicomponent density that is fitted, it is hard to see how one would proceed to solve this problem, even if one could modify the, K-S or Chi-squared goodness of fit procedure to deal with slices or sections. Even if one could find a variant which would apply to sections or slices, in the process of generalizing these two procedures, one would assuredly lose the major advantage that these two methods have over the use of R(x) when the procedures are used in the univariate case. Specifically, it is hard to see how accurate significance levels could be obtained for these generalizations.

Besides their reliance on model-specific estimation procedures, it is also relevant to point out that both the Chi-square goodness of fit and the K-S method are associated with particular choices of nonparametric estimators. Through its dependence on class interval frequencies, the Chi-square method is closely connected with the conventional histogram while K-S methodology is of course dependent upon the sample cumulative step function. In Tarter and Kronmal (1970) it is shown that in terms of the underlying MISE metric, there will always exist a Fourier series density estimator which is superior to its limiting case, which in the case of

unweighted series, happens to be the sample cumulative. Thus, one has good reason to suspect that a model identification method which is based on the sample cumulative can be substantially improved upon by the methods described above.

In the case of a methodology which is related to the conventional histogram, by the extending the same logic which leads the STATGRAF and other Chi-Square goodness-of-fit programs to allow end intervals to be pooled, one would think that methods such as those presented by Scott (1984), which greatly improve the conventional histogram, could be very successfully applied to find improvements of the conventional Chi-square goodness of fit test. Of course one could also use these methods to estimate the f and F arguments of the R(x) functional and in this way circumvent the problem of estimating the parameters of the "fitted" model as a preliminary to checking on the validity of model family choice.

7. Generalizations: There are of course many possible ways of generalizing the methodology presented above. For example, PROPERTY 3 of section 4 implies that if one chooses to graph the derivative of R(x), R'(x), one can select from among members of the class of "reciprocalmodels" by making use of the identity $R'(x) = 2\sigma(x-\alpha)$ where σ and α are defined in section 4.
In the bivariate case, one can combine the procedures described in this paper with those presented in Tarter and Freeman (1987) and obtain a graphical method for distinguishing between two possible departures from the assumptions which underlie linear regression, where regression residuals can have any one of the logmodels, loglogmodels or even recipricalmodels as their distribution. Unlike the logmodel case, we have not found it useful to pursue the above lines of inquiry at this time. Instead, the sensitivity of the R(x) estimates obtained in our GKS-FORTRAN interactive graphical system, clearly suggest one particular area of further fruitful investigation. By forming composites of various available nonparametric density estimators, where each composite is customized for a particular R(x) application, we hope to both facilitate certain areas of data exploration and simultaneously learn more about the advantages and limitations of the estimators upon which the application of the R(x) functional depends.

ACKNOWLEDGEMENTS

Research was supported by U. S. Environmental Protection Agency Grant R813766 and the Health Effects Component of the University of California Toxic Substances Program. The authors would like to thank Drs. E. Margosches and H. Pitcher for their useful suggestions and Ms. C. Mellin for technical assistance.

REFERENCES

Aitchison, J. and Brown, J.A.C., (1957), The Lognormal Distribution, Cambridge University Press, Cambridge.

Cohen, A. C. Jr. (1951), "Estimating Parameters of Logarithmic-Normal Distributions by Maximum Likelihood." Journal of the American Statistical Association, 46, 206-212

Dixon, W. J. and Massey, F. J. Jr. (1983), Introduction to Statistical Analysis (4-th Edition), McGraw Hill, New York.

Elderton, W. P. (1953), Frequency Curves and Correlation (4-th Edition), Harren Press, Washington D.C.

Gross, A. J. and Clark, V. A. (1975), Survival Distributions: Reliability Applications in the Biomedical Sciences, Wiley, New York.

Johnson, N. L.(1949), "Systems of Frequency Curves Generated by Methods of Translation," Biometrika, 36, 149-176.

Johnson, N. L. and Kotz, S. (1970), Continuous Univariate Distributions - 1, Houghton Mifflin, Boston.

Kendall, M. and Stuart, A. (1959), The Advanced Theory of Statistics, Volume 1 - Distribution Theory, Griffin, London.

Kronmal, R. and Tarter, M. (1968), "The Estimation of Probability Densities and Cumulatives by Fourier Series Methods," Journal of the American Statistical Association, 63, 925-952.

Rudin, W., (1953), Principals of Mathematical Analysis, McGraw Hill, New York.

Scott, D.W., (1984), "Multivariate Density Function Representation, "National Computer Graphics ISBN 0-941514-05-06, 794-800.

Statgraphics (1988), User's Guide, Statistical Graphics Corporation, Rockville, Maryland.

Tapia, R.A. and Thompson, J. R. (1978), Nonparametric Probability Density Estimation, The Johns Hopkins University Press, Baltimore.

Tarter, M. (1968), "Inverse Cumulative Approximation and Applications," Biometrika, 55, 29-41.

Tarter, M. (1979), "Biocomputational Methodology, An Adjunct to Theory and Applications," Biometrics, 35, 9-24.

Tarter, M., Cooper, R.C., and Freeman, W.R. (1983), "A Graphical Analysis of the Interrelationships among Waterborne Asbestos, Digestive System Cancer and Population Density," Environmental Health Perspectives, 53, 79-89.

Tarter, M., Kowalski, C. J. (1972), "A New Test for and Class of Transformations to Normality," Technometrics, 14, 735-744.

Tarter, M. and Kronmal R. (1970), "On Multivariate Density Estimates Based on Orthogonal Expansions," The Annals of Mathematical Statistics, 41, 718-722.

Thompson, J.R. Personal Communication.

Tocher, K.D. (1963), The Art of Simulation, Van Nostrand, Princeton.

ROBUSTNESS OF WEIGHTED ESTIMATORS OF LOCATION: A SMALL-SAMPLE STUDY

Gregory Campbell and Richard I. Shrager, National Institutes of Health

The problem of estimation of location is considered in the context of known as well as misspecified weights. For the one-sample problem, the studied estimators include weighted analogs of the mean, the median, the median of the Walsh averages, and Huber M-estimators, as well as a computer-intensive procedure which minimizes the weighted sum of absolute values of pairwise sums and differences of deviations. For estimators which employ a weighted median, interpolation to improve performance is considered. The estimators are evaluated by computer simulation with respect to robustness to weight misspecification as well as robustness to outliers. These simulations, together with the Kantorovich inequalities for bounds on the asymptotic inefficiencies, provide insight concerning the performance of these estimators with misspecified weights.

1. INTRODUCTION

There are many situations in which there is a natural weight associated with each of the observations. (Here the weight refers to a fixed weight attached to each observation as opposed to the W-estimates of Tukey or to the iterative reweighting schemes so useful in the calculation of some estimators.) For example, if each observation is the summary measure of location for a group of data, one might use the inverse of some measure of dispersion for the weight. Or in the regression problem there are cases in which the optimal weights of the pairwise slopes depend on the spacing of the design (independent) variables; see Jaeckel (1972) and Scholz (1978) for details. In proportional sampling situations, it is often the case that the weights are merely the known probabilities associated with the sampling plan.

In the early 1970's, the Princeton Study (Andrews et al., 1972) looked at the question of robustness for the one-sample location problem. Of particular interest in that study was the robustness of the estimators of location to the presence of outliers, and to a lesser extent, to the misspecification of the distribution. In addition to those varieties of robustness, the present study considers the behavior of estimators of location for the situation in which the observations have weights attached to them but the weights are either known and correctly specified or else misspecified. For example, suppose that the observations are weighted but the weights are ignored and an equal-weight estimator is employed. How robust is such an estimator to this misspecification? Also of interest is the robustness to distribution and to outliers. It is impossible to study every estimator; in particular, most estimators which are relatively inefficient but have a very high breakdown constant are not included, in that for contamination rates near 50% it is not clear what is contaminating what. As for robustness to distribution, suppose, for example, that the distribution is thought to be double exponential but really turns out to be normal? Of primary interest is the effect on these estimators in small-sample cases when the weights are misspecified.

2. THE ESTIMATORS

The estimators considered in this paper are as follows:

1. weighted mean (WMEAN)

$$\text{WMEAN} = \frac{\sum w_i^2 X_i}{\sum w_i^2}$$

where, here and throughout the paper, an unlabelled sum runs from 1 to n. This estimator is the weighted least squares estimator for the squared weights. In addition, if the observations are from normal distributions with the same location but different standard deviations σ_i and if the weights are optimally specified ($w_i=1/\sigma_i$), this estimator is not only unbiased but uniform minimum variance unbiased estimator, the maximum likelihood estimator, and asymptotically optimal for the location.

2. weighted median (WMED)

$$\text{WMED} = \text{med}\ \{X_i \text{ with wt } w_i\}$$

This estimator minimizes the weighted sum of absolute deviations. It is median unbiased. If the distribution is double exponential (Laplace), this estimator is maximum likelihood and has maximum efficiency.

3. Huber's weighted M-estimator (WH15)

WH15 is the implicit solution of μ in

$$\sum w_i \psi\left(w_i \frac{X_i-\mu}{\sigma}\right) = 0$$

where

$$\psi(z) = \begin{cases} 1.5 & \text{if } z>1.5 \\ z & \text{if } |z|\le 1.5 \\ -1.5 & \text{if } z<-1.5 \end{cases}$$

and σ is defined iteratively by

$$\sigma = \frac{\text{med}\ \{w_i|x_i-\mu|\}}{.6754},$$

where .6745 is the median of $|Z|$, for Z a standard normal random variable as suggested by Hampel in Andrews et al (1972). This is the weighted analog of Huber's estimator with value c of 1.5, hence the abbreviation WH15. For the equal weights case, it is the maximum likelihood estimate for the least informative distribution and it minimizes the Fisher information (Huber, 1981).

4. weighted pseudo-median (WPMED)

$$\text{WPMED} = \underset{i\le j}{\text{med}} \left\{ \frac{w_i X_i + w_j X_j}{w_i + w_j} \quad \text{with wt.} = w_i + w_j \right\}.$$

This estimator minimizes the absolute value of the pairwise sums given by:

$$\sum_{i\le j} | w_i(x_i-\mu) + w_j(x_j-\mu) |$$

This is the weighted median of the weighted Walsh averages and reduces in the equal weight case to the median of the Walsh averages, the Hodges-Lehmann estimator associated with the Wilcoxon signed rank statistic. As such it relies on the symmetry of the distribution about the unknown parameter. For equal weights, it is an asymptotically efficient R-estimate for the logistic distribution (Huber, 1981).

5. Weighted pairwise Least Absolute Value Sum-Difference (WLAVSD)

WLAVSD is the implicit solution to the minimization of the following function in μ:

$$\sum_{i\le j} | w_i(x_i-\mu) + w_j(x_j-\mu) | + \sum_{i<j} | w_i(x_i-\mu) - w_j(x_j-\mu) |$$

This also minimizes the weighted sum of the ordered absolute residuals. It can be calculated as a weighted least absolute value minimization problem on the n^2 sums and differences, with weights; hence, it is very computationally intensive. In the case of equal weights, WLAVSD reduces to the median of the Walsh averages, and hence can be thought of, along with WPMED, as a generalization of PMED.

These five are the weighted estimators studied here. The estimators corresponding to the equal weights are denoted with the prefix W omitted: MEAN, MED, H15, and PMED. Admittedly there are other appropriate estimators which one might have also included. Of particular interest in this study is the examination of the one-sample behavior of estimators which will be generalizable to the regression setting.

3. SIMULATIONS

Simulations were performed to evaluate the behavior of these various estimators under the correct weights and also under incorrect ones.

Denoting the minimum of the n weights by 1 and the maximum by R (≥1), the following weight schemes were used, where $w_i = 1/\sigma_i$ for the known standard deviations σ_i:

1. Equal -- all weights equal to 1 (R=1).
2. Extreme -- half the weights at 1, half at R.
3. Uniform -- equally spaced, 1 to R.

The following distributions were used to generate pseudo-random numbers using the IMSL statistical routines (IMSL, 1982):

1. Normal (NOR)
2. Contaminated normal (CNOR) -- 20% contamination of a normal distribution with another normal with the same mean and three times the standard deviation.
3. Double exponential (DEXP)
4. Logistic (LGST)
5. Uniform (UNIF)

All were selected to have theoretical variances of 1. One thousand replications of each sample of size n were performed. In order to evaluate estimators, for each sample of size n, the estimator is calculated. For the thousand estimators of the known quantity, the variance is calculated and used to compare estimators.

4. CONSIDERATIONS FOR WEIGHTED MEDIANS

Because several of the estimators involve weighted medians, important choices confront one when considering the small-sample behavior of such estimators. In large samples these considerations are of little importance, in that the estimators are asymptotically equivalent. But the choice is quite important in small samples. The issue is whether or not one should interpolate to obtain the weighted median, and, if so, which one of a number of interpolations to employ. With arbitrary weights or a different number of points, differences become apparent. Define the weighted empirical (sample) distribution function as follows:

$$F_n(x) = \begin{cases} 0 & \text{for } x<x_1 \\ s_j & \text{for } x_j \le x < x_{j+1},\ j=1,\ldots,n-1 \\ 1 & \text{for } x \ge x_n \end{cases}$$

where $s_j = W_j/W_n$ and $W_j = \sum_{i=1}^{j} w_i$. This function is a discrete, step function. It might be expected that a continuous version might outperform it. Consider the following possible interpolation schemes, based on the sample weighted distribution function $F_n(x)$, where without loss of generality, the data are $x_1 \le x_2 \le \ldots \le x_n$ and w_i is the weight corresponding to x_i.

A. simple --- always use $F_n^{-1}(.5)$. If $F_n(x)=.5$ for $x_i \le x < x_{i+1}$, then by convention the simple median is $(x_i+x_{i+1})/2$.

B. mid-data --- using the sample distribution function and plotting only the midpoints in the horizontal line segments, linearize; i. e., plot the points in the table

x	x_1	$\frac{x_1+x_2}{2}$	$\frac{x_2+x_3}{2}$	...	x_n
s	$\frac{s_1}{2}$	s_1	s_2	...	$\frac{s_{n-1}+s_n}{2}$

and connect by line segments. The mid-data median is the unique inverse at .5.

C. mid-weight --- using the sample distribution functions and plotting only the midpoints of the vertical line segments, linearize. In other words, use the table below and connect by line segments.

x	x_1	x_2	x_3	...	x_n
s	$\frac{s_1}{2}$	$\frac{s_1+s_2}{2}$	$\frac{s_2+s_3}{2}$	...	$\frac{s_{n-1}+s_n}{2}$

The mid-weight median is the inverse at .5.

D. mixed --- using the sample distribution function and plotting the midpoints of the horizontal and vertical line segments, linearize; i.e., merge the above two tables

x	x_1	$\frac{x_1+x_2}{2}$	x_2	$\frac{x_2+x_3}{2}$	x_3	...	x_n
s	$\frac{s_1}{2}$	s_1	$\frac{s_1+s_2}{2}$	s_3	$\frac{s_2+s_3}{2}$	...	$\frac{s_{n-1}+s_n}{2}$

and connect by line segments and take the inverse at .5 to obtain the mixed median.

Note in the equal-weight case that the usual median convention is obtained only in the simple and mixed cases and not in general in either the mid-data for odd n nor the mid-weight cases for tied x's. Differences between these two flawed interpolations and the mixed interpolation median are illustrated in Figures 1 and 2. Table 1 reports the variances associated with the simple versus mixed interpolated weighted medians using 1000 simulations for n = 10 and R = 4 for the equally spaced weights. Note that the mixed median variance is 10% smaller than the simple for NOR, CNOR, and DEXP and even more for UNIF. Such superiority of the mixed median cannot be ignored in the calculation of weighted medians and hence in all simulations that follow, the mixed interpolated median is used.

TABLE 1: VARIANCES (x10000) FOR EQUALLY SPACED WEIGHTS (R=4) FOR SIMPLE AND MIXED INTERPOLATED MEDIANS (n=10)

	NOR	CNOR	DEXP	LGST	UNIF
SIMPLE MEDIAN	2142	1222	1153	1699	3427
MIXED MEDIAN	1905	1100	1041	1580	2979

5. COMPARISON OF THE ESTIMATORS

For each sample of size n, all 9 estimators MEAN H15 PMED MED WMEAN WH15 WPMED WLAVSD WMED were calculated. From the 1000 replications, the mean and variance (based on the known location) were calculated and reported in the tables that follow. Note that simulations were carried out for n=10 and n=20. In addition, the asymptotic variances where known are also reported.

Consider Table 2 which reports the variances of the estimators in the equal weights case. Note for this unweighted case that the MEAN performs quite well for the distributions NOR and UNIF. Furthermore, the variances for MEAN do not vary much for the other distributions. For the heavy-tailed distributions CNOR, DEXP, and LGST, the estimators MED and PMED appear to perform quite well, with MED doing slightly better for DEXP and PMED for LGST and CNOR. Comparison of the behavior of the estimators for n=10 and n=20 and the asymptotic variance is instructive in evaluating the differences between the small-sample and the asymptotic performance of these estimators. In Table 2, the approximate standard deviations associated with the mean for the distributions NOR, CNOR, DEXP, LGST, and UNIF depend on the kurtosis of the distribution; they are (x1000n) 45, 66, 70, 57, and 28 respectively. Note that these standard deviations values do not depend on n, and, in general, cannot be expected to converge to the reported asymptotic variance as n, the size of the sample, tends to infinity. It is noteworthy that small-sample variances for MED are quite different from the asymptotic variance, and to a lesser extent for PMED. For the medians and, to a lesser extent, the pseudo-medians, it is conjectured that as n increases the sample variances associated with the replications tend to decrease to values that vary about the asymptotic variances reported in the table. Also included in this table are the results of the simulations reported by Andrews et al (1972). A quick glance confirms that similar results are obtained here as in their study. For a related study of the small-sample behavior of the associated tests for the estimators MEAN, MED, and PMED for the distributions NOR, DEXP, LGST, UNIF, and the Cauchy, there is the report of the empirical powers in Randles and Wolfe (1979).

Consider Table 3 in which, for extreme weights and R=4, the variances are presented for the unweighted and weighted estimators. Among the weighted estimators (and hence all estimators since each weighted estimator outperforms its unweighted analog), note that WMEAN is best for NOR and UNIF, that WMED is best for DEXP, and that WLAVSD and, to a lesser extent, WPMED perform well for LGST, as one might expect from Table 2. If attention is turned to the estimators which ignore the weights (the estimators without the prefix W), then MEAN is very poorly behaved, even for the NOR distribution. Also, MED is uniformly best of the four unweighted estimators for each of the distributions. The asymptotic variance rows are obtained using the asymptotic distribution for the weighted estimator based on a fixed number, n, of weights and letting the number of observations, k, at each weight tend to infinity and multiplying the asymptotic variance by 10000n. Note that for this extreme weight case this asymptotic variance result does not depend on n. As in Table 2, MED and now WMED have

small-sample variances which are much larger than the asymptotic ones for DEXP.

For Table 4, for equally-spaced weights and R=4, note that the variances are generally smaller than observed in the extreme weight case for the same R in Table 3, as one might have expected. Among the weighted estimators, WMEAN is best for NOR and UNIF and poorly behaved for DEXP and CNOR. WMED performs fairly well for DEXP for n=10 and n=20, although not as well as the asymptotic result might have one believe. Note that in this case of equally spaced weights the asymptotic variances are different only for WMED for n=10 versus n=20. WLAVSD performs well for LGST, CNOR and DEXP. It is no surprise that WMEAN is not robust to the heavy-tailed distributions whereas both WMED, WLAVSD and WPMED are. Among the estimators which ignore the weights, MED does well for NOR, DEXP, LGST and PMED performs well for NOR and UNIF. As in Table 3, the MEAN is dismally behaved for the misspecified weights. It is interesting for UNIF that WMED has larger variance than PMED and H15.

6. KANTOROVICH'S INEQUALITY

Cressie (1980) considered weighted M-estimation and its large-sample behavior relative to weight misspecification. As mentioned there, for known weights w_i, Kantorovich (1948) proved that the measure, r, of inefficiency, given by the ratio of the variances of the unweighted MEAN to the optimally weighted WMEAN, is bounded by:

$$r(\text{MEAN},\text{WMEAN}) \leq 1 + \frac{(R^2-1)^2}{4R^2} \qquad (1)$$

where $R = M/m$, $M = \max\{w_i\sigma_i\}$ and $m = \min\{w_i\sigma_i\}$. For the median, Tukey has shown, as mentioned by Cressie, that

$$r(\text{MED},\text{WMED}) \leq 1 + \frac{(R-1)^2}{4R} \qquad (2)$$

where R is as above. (Cressie also reports a Kantorovich inequality for M-estimates, but it is inappropriate here in that the weighted M-estimates considered by him are either in the homogeneous variance case or have a psi function which is homogeneous with regard to internal weights and unfortunately the influence function for the H15 is not homogeneous.)

For R = 4, the Kantorovich bounds in equations (1) and (2) are 4.516 and 1.5625. These can be compared with the small-sample inefficiencies as observed by the ratios of the estimated variances of Tables 3 and 4. These ratios are reported in Table 5 (x1000). As is no surprise, the small-sample inefficiency ratios are larger for the extreme weights than for the equally-spaced ones; in fact it is exactly the extreme weights that give the bound for the inequality. It is interesting to note that while MED has the best observed efficiencies across the five distributions, PMED is somewhat competitive, especially in the equally spaced weighting situation. The choice of which estimator to use clearly depends upon the quantification of the likely weight misspecification as well as the possibility of outliers or deviation from believed distribution. For small R, weights play a less important role, and for large R, the possibility of misspecification becomes crucial in the selection of a one-sample estimator. As advocated by Tukey and as mentioned by Cressie (1980) for R<2 the mean has some advantage and only for larger R does the median assert itself. This is certainly seen in this study for R=4 with these two weighting schemes. Note for equally spaced weights and R=4 that PMED competes quite well with MED when one takes into account the poor efficiency behavior of the MED to PMED for NOR, CNOR, LGST and UNIF (recall the asymptotic relative efficiencies of MED to PMED for these four distributions are 2/3, .79, 3/4 and 1/3, respectively).

7. CONCLUSIONS

Weights, even if hidden in an estimation problem, can play an important role in the selection of the estimator. Failure to recognize the existence of unequal weights can be disastrous even if the distribution is correctly identified. In particular, the mean is terribly non-robust with respect to weight misspecification, even with data that are normal. The median, on the other hand, while robust with regard to weights and with regard to outliers, does have some poor efficiency properties if the distribution is not heavy-tailed. A compromise might be to use the Hodges-Lehmann, pseudo-median estimator (whose weighted analogs are WLAVSD and WPMED), which has reasonable robustness to outliers and distribution and which is more robust to weight misspecification than either the mean or the Huber M-estimate.

The small-sample behavior differs from the known asymptotic results for several of the studied situations. In particular, one surprising result is the margin by which the small-sample estimated variance differs in the case of the median from the asymptotic prediction. This substantial discrepancy is expected to widen if one were to examine related estimators in the simple linear regression problem. It is also interesting that there is a difference in the other direction with regard to the normal and uniform distributions; namely, that the observed small-sample variances are smaller than expected from the asymptotic prediction. These results may not seem so surprising if for the median one recalls that the asymptotic approximation is influenced by the smoothness of the density in the neighborhood of the theoretical median.

In conclusion, it appears that there is no optimal resolution concerning the selection of an estimator that is robust with respect to weights as well as to outliers. The choice of an estimator depends upon the weights, their spacing and range R (even if not identified), and the behavior of the distribution in the tails.

The authors gratefully acknowledge the technical and editorial assistance of M. Hodges of the Division of Computer Research and Technology.

REFERENCES

Andrews, D.F., Bickel, P.J., Hampel, F.R., Huber, P.J., Rogers, W.H., and Tukey, J.W. (1972). *Robust Estimates of Location: Survey and Advances*. Princeton University Press: Princeton, NJ.

Cressie, N. (1980). Weighted M-estimation in the presence of unequal scale. *Statistica Neerlandica 34*, 19-32.

Huber, P. J. (1981). *Robust Statistics*. John Wiley and Sons: New York, New York.

IMSL (1982). *IMSL Library*. IMSL: Houston, Texas.

Jaeckel, L. (1972). Estimating regression coefficients by minimizing the dispersion of the residuals. *Ann. Math. Statist. 43*, 1449-1458.

Kantorovich, L. (1948). Functional analysis and applied mathematics. *Usphehi Mat. Nauk. 3*, 89-185.

Randles, R. H. and Wolfe, D. A. (1979). *Introduction to the Theory of Nonparametric Statistics*. John Wiley and Sons: New York, New York.

Scholz, F.-W. (1978). Weighted median regression estimates. *Ann. Statist. 6*, 603-609.

TABLE 2: VARIANCES (x1000n) - EQUAL WTS

	NOR	CNOR	DEXP	LGST	UNIF
MEAN					
n=10	1011	978	958	1095	1038
n=20	1013	1029	1112	1009	993
PRINC*	1000	1038	1050		
ASYMP	1000	1000	1000	1000	1000
H15					
n=10	1091	735	765	1014	1177
n=20	1071	678	881	946	1102
PRINC*	1031	690	820		
PMED					
n=10	1121	706	723	1039	1300
n=20	1077	662	823	938	1205
PRINC*	1063	673	745		
ASYMP	1047	635	667	912	1000
MED					
n=10	1476	723	723	1183	2329
n=20	1515	735	743	1144	2612
PRINC*	1366	708	685		
ASYMP	1571	804	500	1216	3000

*Values as reported in the Princeton study by Andrews et al (1972): n=10 for NOR and CNOR, n=20 for DEXP.

TABLE 3: VARIANCES (x10000n)--EXTREME WTS (R=4)

		NOR	CNOR	DEXP	LGST	UNIF
MEAN	(n=10)	5471	5084	5467	5892	5765
	(n=20)	5502	5092	5259	5491	5174
	(asym)	5313	5313	5313	5313	5313
H15	(n=10)	4145	2568	2919	3709	5438
	(n=20)	3688	2103	2288	3146	4955
PMED	(n=10)	3581	2149	2503	3334	5041
	(n=20)	3159	1737	1950	2742	4101
MED	(n=10)	2452	1488	1447	2141	3654
	(n=20)	2478	1295	1294	1955	3760
	(asym)	2513	1287	800	1945	4800
WMEAN	(n=10)	1155	1190	1067	1266	1155
	(n=20)	1235	1157	1228	1194	1228
	(asym)	1177	1177	1177	1177	1177
WH15	(n=10)	1209	890	875	1224	1274
	(n=20)	1296	803	975	1132	1327
WPMED	(n=10)	1323	913	895	1280	1489
	(n=20)	1367	826	965	1215	1523
WLAVSD	(n=10)	1225	864	849	1218	1363
	(n=20)	1294	784	933	1133	1415
WMED	(n=10)	1589	898	861	1483	2361
	(n=20)	1815	899	921	1359	2797
	(asym)	1848	946	588	1430	3530

TABLE 4: VARIANCES (x10000n)--EQUALLY SPACED WTS (R=4)

		NOR	CNOR	DEXP	LGST	UNIF
MEAN	(n=10)	2863	2679	2824	3103	2991
	(n=20)	2711	2487	2765	2742	2684
	(asym)	2836	2836	2836	2836	2836
H15	(n=10)	2379	1582	1648	2143	2806
	(n=20)	2135	1373	1563	1976	2616
PMED	(n=10)	2299	1461	1517	2070	2880
	(n=20)	2101	1292	1389	1903	2702
MED	(n=10)	2392	1283	1238	1964	3599
	(n=20)	2421	1281	1052	1996	4167
	(asym)	2513	1287	800	1945	4800
WMEAN	(n=10)	1394	1421	1289	1507	1396
	(n=20)	1351	1432	1488	1378	1360
	(asym)	1395	1395	1395	1395	1395
WH15	(n=10)	1467	1049	1038	1427	1555
	(n=20)	1395	979	1176	1330	1479
WPMED	(n=10)	1556	1047	1044	1463	1784
	(n=20)	1492	986	1107	1369	1684
WLAVSD	(n=10)	1510	1039	996	1423	1670
	(n=20)	1452	973	1084	1313	1586
WMED	(n=10)	1978	1089	1001	1661	2943
	(asym,n=10)	2192	1122	698	1696	4186
	(n=20)	2044	1110	1002	1625	3516
	(asym,n=20)	2219	1136	706	1718	4238

TABLE 5: OBSERVED MEASURE OF INEFFICIENCY (x1000) (n=10)

	NOR	CNOR	DEXP	LGST	UNIF
EXTREME WTS (R=4)					
MEAN	4739	4271	5124	4652	4993
H15	3429	2887	3335	3031	4269
PMED	2708	2353	2797	2605	3386
MED	1544	1656	1681	1444	1547
EQUALLY SPACED WTS (R=4)					
MEAN	2053	1885	2191	2059	2143
H15	1622	1509	1588	1502	1805
PMED	1477	1396	1453	1415	1614
MED	1209	1178	1238	1183	1223

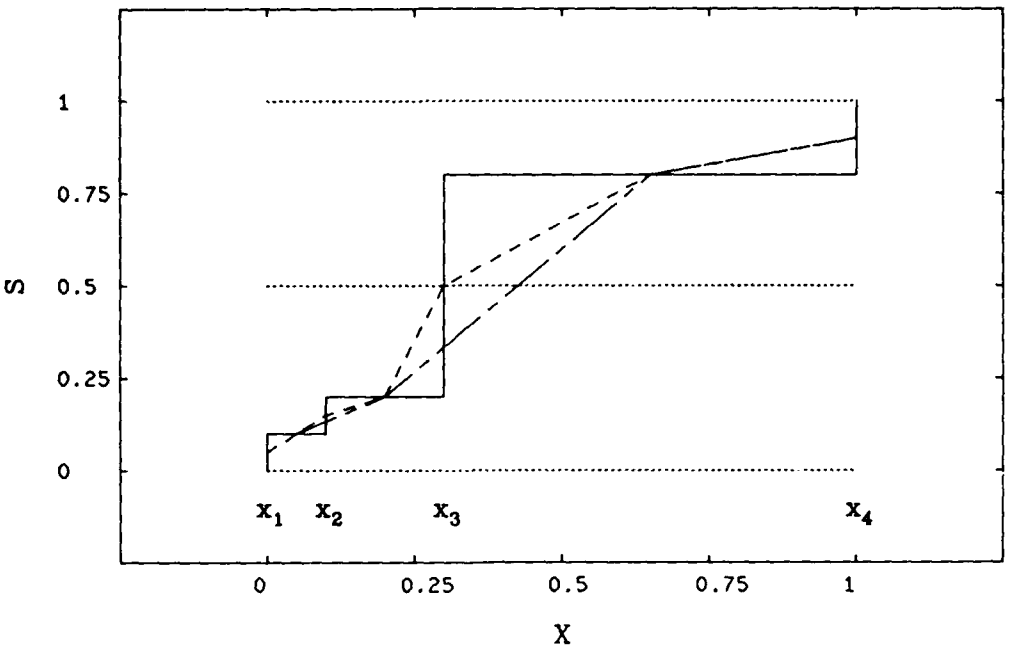


Figure 1. An example showing mid-data (- —— - - - - —— -) and mixed (- - - - - - - - - - -) interpolated medians for n = 4. When $w_1 = w_2 = w_4$, the mixed median is always x_3 (where the short dashed line crosses s = 0.5), bu the mid-data median depends on the spacing of the x's.

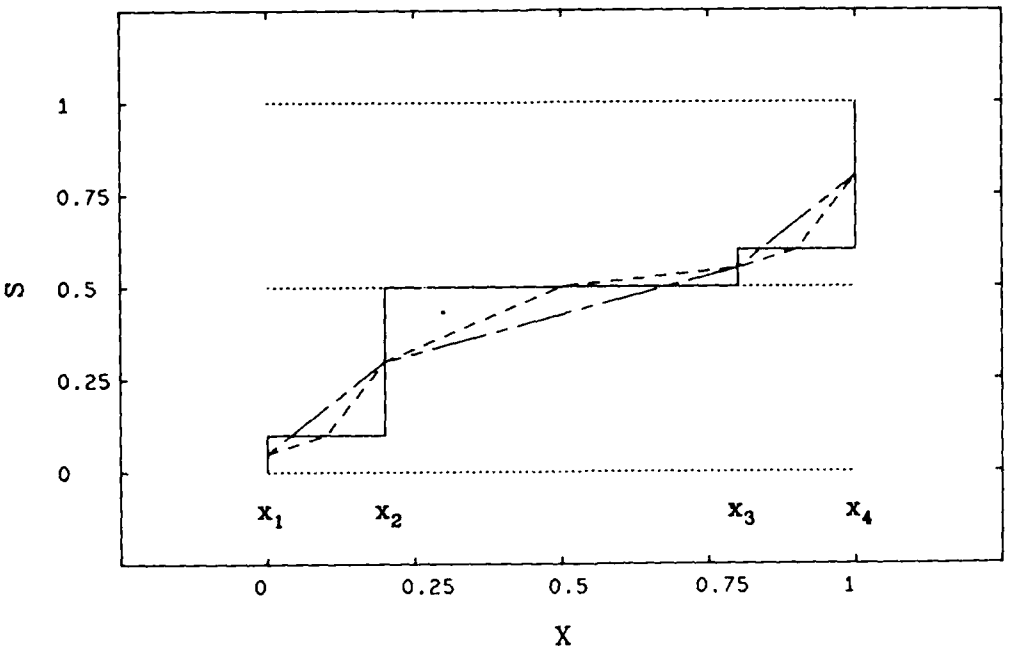


Figure 2. An example showing mid-weight ( ) and mixed (- - - - - - - - - -) interpolated medians for n=4. When $w_1+w_2 = w_3+w_4$, the mixed median is always $(x_2+x_3)/2$ (where the short dashed line crosses s = 0.5), but the mid-weight median depends on the spacing of the s's.

APPROXIMATIONS OF THE WILCOXON RANK SUM TEST IN SMALL SAMPLES WITH LOTS OF TIES

Arthur R. Silverberg, Food and Drug Administration

Abstract

The Wilcoxon-Mann-Whitney rank sum test for two independent samples is frequently used with data having ties. Although there are computer programs to calculate the exact randomization test, even for small samples, computer packages use approximations based upon the normal, Student's t distribution, or the distribution without ties. For each of the small sample sizes considered in this paper, all distributions of obtaining ties were considered, as well as all permutations of the ordering of the ties. The exact distribution with ties was compared to, the tabulated value without ties, normal approximations with and without continuity corrections, and Edgeworth expansions with and without continuity corrections. The purpose of looking at all these distributions was to quantify the accuracy of the common approximations, rather than to develop any new approximations.

1. Introduction

Recommendations for approximations to the Wilcoxon-Mann-Whitney rank sum statistic in the case of ties vary. Conover [2] suggests using the normal approximation with no continuity correction when there are ties. Klotz [6] found that the Edgeworth approximation gave only a small improvement over the normal approximation, both approximations used a continuity correction. Hollander and Wolfe [4] suggest the use of the exact tables for no ties, when ties are present and the samples are small. When the samples are large, the normal approximation with no continuity correction is suggested. When the largest proportion of sample values in a tied category is not close to 1, Lehmann [7] suggests the use of the continuity corrected normal approximation. Emerson and Moses [3] recommend using exact calculations unless both sample sizes are at least 10. They state that the normal approximation is unreliable when over half the observations fall into one category. Although Emerson and Moses state that the use of the continuity correction makes little difference, they recommend against the use of the continuity correction because of the unequal spacing of the statistic.

Klotz [6] provides an algorithm and flow chart for calculating the exact probability of the Wilcoxon-Mann-Whitney statistic. A network algorithm for the Exact Wilcoxon-Mann-Whitney test with ties is found in Mehta, Patel and Tsiatis [9]. This paper contains some typographic errors. A fuller explanation of the network algorithm, in the case of 2xk contingency tables, is found in Mehta and Patel [8].

Major computer packages use a variety of approximations. BMDP [1] uses a normal approximation. It is not clear from the doumentation if a continuity correction is used. SAS [10] provides the the normal approximation with and without continuity correction, and a t-distribution approximation based on n_1+n_2-1 degrees of freedom. SPSS-X [12] gives a normal approximation with no continuity correction and for $n_1+n_2<30$ the tabular p-value from a table assuming no ties. IMSL [5] uses three normal approximations without the continuity correction. Ties are broken to give both the highest nd lowest possible statistics. The approximtion is then applied to these two statistics as well as the original data with ties.

This author has written a computer program to calculate the exact value of the Wilcoxon-Mann-Whitney statistic with ties for IBM-PC compatible computer based upon the Mehta, Patel and Tsiatis algorithm. The program was written in compiled Turbo Pascal Version 3.0 and is available to interested parties who mail a formatted diskette (either 3.5 or 5.25 inch) in a self-addressed mailer to the author.

2. Exact Distribution Under H_o

Let us denote the smaller sample size by n_1 and the larger sample size by n_2, $n_1+n_2=N$. Let $W=\sum_{i=1}^{n_1} R_i$ when R_i is the rank of observation i from the smaller sample and let t_j $j=1\cdots c$ be the number of observations from both samples that are tied in category j, $R_j<R_{j+1}$. It is well known that

$$\mu=n_1\frac{(N+1)}{2}$$

and

$$\sigma^2=\frac{n_1n_2}{12}\left[N+1-\frac{\sum_{j=1}^{c} t_j(t_j^2-1)}{N(N-1)}\right]$$

It can be shown that

$$\mu_3=\frac{n_1n_2(n_1-n_2)}{4N(N-1)(N-2)}\left[\sum_{j=1}^{c}(R_j-\frac{N+1}{2})t_j(t_j^2-1)\right]$$

This formula for μ_3 seems new. It is easy to see that $\mu_3=0$ when $n_1=n_2$ or if $t_j=t_{c+1-j}$ as pointed out by Klotz [6].

The formula for μ_4 as given at the bottom of the next page seems to be new.

For each total sample size considered N, all distributions of ties were considered. The number of such distributions, p(N) is the number of unrestricted partitions of N. For example for N=5, p(5)=7 since

1) 1+1+1+1+1=5
2) 1+1+1+2=5
3) 1+1+3=5
4) 1+2+2=5
5) 1+4=5
6) 2+3=5
7) 5=5

For small N,

N	4	5	6	7	8	9	10	11
p(n)	5	7	11	15	22	30	42	56

N	12	13	14	15	16	17	18	19
p(n)	77	101	135	176	231	297	385	490

Ties may be in any order. So for each distribution of ties, we considered all $c!/((\#t_i=1)! \times \cdots \times (\#t_i=N))$ permutations of the ties. For example $t_1=t_2=1$ and $t_3=3$ therefore, $3!/(2! \times 0! \times 1! \times 0! \times 0!)=3$.

	$R_1x(1)$	$R_2x(2)$	$R_4x(3)$	$R_4x(4)$	$R_4x(5)$
1)	1	3	3	3	5
2)	2	2	2	4	5

The Wilcoxon-Mann-Whitney rank sum statistic was computed for all possible samples of size $n_1 \leq n_2$. The exact probabilities were compared to a number of approximations explained below when either the exact value or the respective approximation was less than .1.

3. Approximations

All comparisons were for one-sided tests and p-values were for the alternative hypothesis of the location of population one smaller than that of population two.

Two approximations were based upon the standard Tables that assume no ties, and are denoted as approximations T_1 and T_2. For approximation T_1, look up the statistic in the Table and find the p-value corresponding to that integer, or the next larger value if the statistic is non-integral (it must always be either an integer or an integer plus one-half) because of ties. T_2 is the same as T_1 except that linear interpolation is performed for non-integral values.

Three approximations were based upon the standard Normal distribution, and are denoted as approximations N_0, N_1 and N_{gcd}. Approximation N_0 is the normal approximation with no continuity correction. Approximation N_1 is the normal approximation with the continuity correction based upon assuming the lattice spacing $\lambda=1$. Smid [11] has shown that the spacings of the exact distribution of the Wilcoxon-Mann-Whitney statistic is a multiple of the greatest common divisor (gcd) of $.5 \times \text{gcd}\{t_1+t_2,\cdots,t_{c-1}+t_c\}$. Therefore taking the suggestion of Klotz [6] we use $.25 \times \text{gcd}\{t_1+t_2,\cdots,t_{c-1}\}$ as a continuity correction in N_{gcd}. We have the following relationships among these approximations, $N_0<N_1$, $N_0<N_{gcd}$ and $N_1<=>N_{gcd}$ depending on whether $1<=>.5 \times \text{gcd}\{t_1+t_2,\cdots,t_{c-1}+t_c\}$.

Three approximations were based upon the Edgeworth approximation, E_0, E_1 and E_{gcd} defined in an analogous way to N_0, N_1 and N_{gcd}. Therefore, $E_0<E_1$, $E_0<E_{gcd}$ and $E_1<=>E_{gcd}$ depending on whether $1<=>.5 \times \text{gcd}\{t_1+t_2,\cdots,t_{c-1}+t_c\}$.

The Edgeworth approximation is given by

$$E(x)=\Phi(x)-(\kappa_3/(6\kappa_2^{1.5}))(x^2-1)Z(x)$$
$$-(\kappa_4/(24\kappa_2^{2}))(x^3-3x)Z(x)$$
$$-(\kappa_3^{2}/(72\kappa_2^{3}))(x^5-10x^3+15x)Z(x)$$

$\Phi(x)$ and $Z(x)$ are the normal cumulative distribution function and the probability distribution respectively, and κ_i are the cumulants.

4. Accuracy of Approximations

The tables give the maximum absolute error |(estimated-actual)|, and relative error |((estimated-actual)/actual)|. The sign is given or $\pm$ when the largest positive error equals the largest negative error. A value 0 means that either all probabilities were estimated correctly or, there were no estimated and no exact p-values less than 0.1. Computations were performed on a VAX 8530 using a Basic language compiler at the Food and Drug Administration.

Table 1A show the largest absolute error over all partitions when either the true or estimated p-value is less thn or equal to 0.1. Even for the sample sizes that are not too small, the largest errors are large. The approximations based upon the standard tables are conservative in that the largest absolute errors tend to occur when estimated>actual.

$$\mu_4=n_1n_2\frac{(N+1)}{240}[n_1n_2(5N+7)-2N(N+1)]$$

$$-\frac{n_1n_2[N(N+1)-6n_1n_2]}{240N(N-1)(N-2)(N-3)} \times \sum_{j=1}^{c} t_j(t_j^2-1)(120R_j(R_j-(N+1))+3t_j^2)$$

$$+\frac{n_1n_2\sum_{j=1}^{c} t_j(t_j^2-1)}{240N(N-1)(N-2)(N-3)} \times [5(n_1-1)(n_2-1)\sum_{j=1}^{c} t_j(t_j^2-1)$$

$$-42n_1n_2-10n_1n_2(N+1)(N^2-19N-18)-N(N+1)(20N^2+80N+13)$$

The normal and Edgeworth approximations are not conservative for absolute errors since the largest absolute errors tend to occur when estimated<actual. The best approximation in terms of overall absolute error seems to be, T_2 when $n_1=n_2$, T_1 when $n_1<n_2$ $n_1=1,2$, N_{gcd} when $n_1<n_2$ $n_1=3,4,5$, and E_1 when $n_1<n_2$ $n_1\geq 6$. Approximation N_0 gives the poorest performance in terms of overall absolute error.

Table 1B shows the largest relative error over all partitions when either the true or estimated p-value is less than or equal to 0.1. The largest relative error tends to occur when the actual p-value is smallest, as opposed to the absolute error that does not have this tendency. The larger sample sizes tend to have larger relative overall errors. All approximations studied are conservative in that the largest relative errors tend to occur when estimated>actual for $n_1\geq 3$. The best approximation in terms of overall relative error seems to be, T_1 for $n_1=1,2$, N_0 for $n_1=3,4$, and E_0 for $n_1\geq 5$. The approximations that gives the poorest performance in terms of overall relative error seems to be N_0 for $n_1=1,2$, and T_1 and T_2 for $n_1\geq 3$.

Table 2 shows the largest absolute and relative errors for the partition when there are no ties and the true or estimated p-value is less than or equal to 0.1. The table look-ups of course have zero error so are excluded.

The first four columns of Table 2 contain the absolute errors over the no ties partition when the true or estimated p-value is less than or equal to 0.1. The largest error of the no ties partition is usually at least an order of magnitude smalller than the errors shown in Table 1A. The two normal approximations, and approximation E_0 tend not to be conservative in that the largest errors occur when estimated<actual. Approximation $E_1=E_{gcd}$ tends to be conservative, the largest errors occur when estimated>actual, for the larger sample sizes given. Approximation $E_1=E_{gcd}$ is the better of the two normal and two Edgeworth approximations in terms of largest absolute error. Appproximation N_0 gives the poorest performance in terms of largest absolute error over the no ties partition.

The last four columns of Table 2 contain the largest relative errors of the no ties partition when the true or estimated p-value is less than or equal to 0.1. The largest error of the no ties parititon is frequently nearly as large as the errors shown in Table 1B and in fact in some cases the largest error in Table 1B was from the no ties partition. The two normal approximations tend not to be conservative for larger samples in that the largest relative errors occur when estimated<actual. Approximation E_0 tends to be conservative in that the largest relative errors occur when estimated>actual. Approximation $E_1=E_{gcd}$ does not seem to be easily classifiable as being either conservative or not conservative in terms of largest relative error for the no ties partition. The Edgeworth approximations tend to be better than the normal approximations in terms of largest relative error, with $E_1=E_{gcd}$ usually better than E_0. Approximations N_0 gives the poorest performance for $n_1=1$ in terms of largest relative error of the no ties partition, while approximation $N_1=N_{gcd}$ gives the poorest performance for $n_1\geq 2$.

5. Recommendations

If there are no ties, the Edgeworth approximation with continuity correction, $E_1=E_{gcd}$ is recommended when tables are not available. Since none of the approximations are accurate when there are ties, even for moderate size samples, the calculation of the exact probability of the Wilcoxon-Mann-Whitney statistic is recommended.

References

[1] BMDP Statistical Software Manual (1985). "Subprogram P3S," Berkeley, California.

[2] Conover, W. J. (1980). Practical Nonparametric Statistics, New York: John Wiley & Sons.

[3] Emerson, John D. and Moses, Lincoln E. (1985). "A Note on the Wilcoxon-Mann-Whitney Test for 2 x k Ordered Tables," Biometrics, 41, 303-309.

[4] Hollander, Myles and Wolfe, Douglas A. (1973). Nonparametric Statistical Methods, New York: John Wiley & Sons.

[5] IMSL User's Manual (1987). "Subroutine RNKSM/DRNKSM," Houston, Texas.

[6] Klotz, J. H. (1966). "The Wilcoxon, Ties, and the Computer," Journal of the American Statistical Association, 61, 772-787, corrigenda, 62 (1967), 1520-1521.

[7] Lehmann, E. J. (1975). Nonparametrics: Statistical Methods Based On Ranks, San Francisco: Holden-Day, Inc.

[8] Mehta, Cyrus R. and Patel, Nitin R. (1980). "A Network Algorithm for the Exact Treatment of 2xk Contingency Table," Communications in Statistics, Series B9(6), 649-664.

[9] Mehta, Cyrus R., Patel Nitin R. and Tsiatis, Anastasios A. (1984). "Exact Significance Testing to Establish Treatment Equivalence with Ordered Categorical Data," Biometrics, 40, 819-825.

[10] SAS User Guide: Statistics (1985). "Proc NPAR1WAY," Cary, North Carolina.

[11] Smid, L. J. (1955). "On the Distribution of the Test Statistics of Kendall and Wilcoxon's Two Sample Test When Ties are Present," Statistica Neerlandica, 10, 205-214.

[12] SPSS-X SPSS Statistical Algorithms (1986). "M-W Subcommand of the NPAR Tests," Chicago.

Table 1A: Signed(Max|Estimated-Actual|) All Partitions of Ties

n_1+n_2	T_1	T_2	N_0	N_1	N_{gcd}	E_0	E_1	E_{gcd}
				$n_1=2$				
8	-.036	-.036	-.208	-.176	-.129	-.191	-.116	-.125
9	±.028	±.028	-.192	-.168	-.103	-.185	-.147	-.101
10	∓.044	∓.044	-.177	-.160	-.137	-.166	-.145	-.134
11	-.036	-.036	-.186	-.152	-.110	-.180	-.141	-.108
12	-.060	-.060	-.226	-.145	-.143	-.153	-.134	-.139
13	-.051	-.051	-.217	-.137	-.123	-.177	-.125	-.120
14	-.066	-.066	-.208	-.187	-.162	-.187	-.137	-.157
15	+.057	+.057	-.200	-.182	-.168	-.181	-.159	-.152
16	-.075	-.075	-.202	-.178	-.168	-.176	-.157	-.148
17	-.066	-.066	-.186	-.173	-.165	-.174	-.155	-.143
18	+.065	+.065	-.218	-.169	-.162	-.166	-.153	-.147
				$n_1=3$				
8	+.107	+.107	-.277	-.099	-.107	-.170	-.099	-.108
9	+.190	+.190	-.255	+.137	-.143	-.163	+.151	-.142
10	+.192	+.192	-.237	-.205	+.113	-.212	+.146	+.127
11	+.194	+.194	-.221	-.196	-.126	-.201	+.138	-.128
12	+.195	+.195	-.208	-.188	-.115	-.191	-.163	+.115
13	+.196	+.196	-.197	-.181	-.118	-.183	-.161	-.121
14	+.195	+.195	-.187	-.173	-.140	-.174	-.157	-.141
15	+.196	+.196	-.177	-.167	+.120	-.172	-.153	+.125
16	+.195	+.195	-.169	-.160	-.130	-.158	-.149	-.132
17	+.194	+.194	-.161	-.154	-.146	-.159	-.144	-.147
18	+.194	+.194	-.218	-.148	-.124	-.174	-.137	-.126
				$n_1=4$				
8	+.100	+.100	-.193	-.124	-.136	-.148	-.116	-.130
9	+.095	+.095	-.190	-.103	-.101	-.185	-.095	-.094
10	+.138	+.107	-.181	-.096	-.102	-.170	-.097	-.103
11	+.121	+.121	-.271	-.122	-.129	-.165	-.122	-.130
12	+.194	+.194	-.255	-.147	-.155	-.168	+.139	-.155
13	+.225	+.200	-.241	-.216	+.110	-.215	+.137	+.122
14	+.204	+.204	-.229	-.208	-.126	-.206	+.132	-.130
15	+.229	+.208	-.218	-.201	-.145	-.198	-.175	-.147
16	+.210	+.210	-.208	-.194	-.108	-.191	-.171	-.113
17	+.230	+.212	-.200	-.187	-.125	-.185	-.168	-.129
18	+.213	+.213	-.192	-.181	-.140	-.178	-.164	-.144
				$n_1=5$				
10	+.127	+.099	-.188	-.134	-.144	-.151	-.126	-.137
11	+.104	+.104	-.186	-.115	-.114	-.180	-.107	-.106
12	+.126	+.107	-.180	-.100	-.106	-.169	-.095	-.100
13	+.124	+.124	-.175	-.115	-.121	-.165	-.116	-.122
14	+.136	+.136	-.267	-.153	-.142	-.162	-.135	-.142
15	+.202	+.202	-.255	-.149	+.107	-.172	-.139	+.120
16	+.208	+.208	-.243	-.224	-.116	-.217	-.137	+.119
17	+.214	+.214	-.233	-.216	-.125	-.209	-.135	-.129
18	+.218	+.218	-.224	-.209	-.141	-.203	-.182	-.144
				$n_1=6$				
12	+.106	+.106	-.185	-.140	-.149	-.153	-.132	-.142
13	+.113	+.113	-.184	-.123	-.122	-.177	-.115	-.114
14	+.118	+.118	-.179	-.109	-.115	-.169	-.102	-.108
15	+.129	+.129	-.176	-.159	-.115	-.166	-.111	-.116
16	+.157	+.141	-.277	-.157	-.133	-.163	-.128	-.134
17	+.150	+.150	-.265	-.153	-.150	-.160	-.144	-.150
18	+.210	+.210	-.255	-.236	+.105	-.174	-.142	+.117
				$n_1=7$				
14	+.132	+.113	-.183	-.145	-.152	-.171	-.137	-.145
15	+.122	+.122	-.182	-.129	-.129	-.175	-.121	-.121
16	+.141	+.127	-.202	-.164	-.121	-.168	-.109	-.114
17	+.130	+.130	-.176	-.162	-.110	-.166	-.108	-.112
18	+.145	+.145	-.172	-.159	-.126	-.163	-.122	-.127
				$n_1=8$				
16	+.121	+.121	-.181	-.148	-.155	-.170	-.140	-.147
17	+.129	+.129	-.180	-.166	-.166	-.174	-.126	-.125
18	+.136	+.136	-.198	-.165	-.127	-.168	-.114	-.120
				$n_1=9$				
18	+.096	+.092	-.180	-.167	-.157	-.169	-.142	-.149

Table 1B: Signed(Max(|Estimated-Actual|/Actual)) All Partitions of Ties

n_1+n_2	T_1	T_2	N_0	N_1	N_{gcd}	E_0	E_1	E_{gcd}
				$n_1=2$				
8	-0.33	-0.33	-0.89	-0.79	-0.83	-0.81	-0.68	-0.73
9	±0.33	±0.33	-0.92	-0.84	-0.84	-0.80	-0.69	-0.72
10	±0.33	±0.33	-0.94	-0.89	-0.91	-0.83	-0.72	-0.74
11	±0.33	±0.33	-0.96	-0.92	-0.92	-0.84	-0.77	-0.73
12	±0.50	±0.50	-0.97	-0.95	-0.96	-0.84	-0.80	+0.81
13	+0.50	+0.50	-0.98	-0.96	-0.96	-0.83	-0.81	-0.75
14	+0.50	+0.50	-0.99	-0.97	+1.25	-0.83	-0.80	+1.10
15	+0.60	+0.60	-0.99	-0.98	+1.26	-0.85	-0.78	+0.96
16	+0.60	+0.60	-0.99	-0.99	+1.67	-0.88	-0.77	+1.39
17	+0.60	+0.60	-1.00	+1.08	+1.61	-0.92	-0.81	+1.44
18	+0.67	+0.67	-1.00	+1.25	+2.12	-0.95	-0.86	+1.74
				$n_1=3$				
8	+1.20	+1.20	-0.77	+0.89	-0.70	-0.80	+1.03	+0.80
9	+2.29	+2.29	+1.13	+1.64	+1.38	+1.30	+1.82	+1.55
10	+2.88	+2.88	+1.43	+1.99	+1.70	+1.63	+2.18	+1.90
11	+3.56	+3.56	+1.72	+2.32	+2.01	+1.94	+2.53	+2.23
12	+4.30	+4.30	+2.01	+2.63	+2.31	+2.23	+2.86	+2.54
13	+5.09	+5.09	+2.27	+2.93	+2.59	+2.51	+3.17	+2.83
14	+5.92	+5.92	+2.53	+3.21	+2.86	+2.77	+3.45	+3.10
15	+6.85	+6.85	+2.76	+3.47	+3.10	+3.01	+3.71	+3.35
16	+7.79	+7.79	+2.98	+3.71	+3.33	+3.24	+3.95	+3.59
17	+8.80	+8.80	+3.25	+3.95	+3.59	+3.45	+4.18	+3.81
18	+9.88	+9.88	+3.53	+4.24	+4.04	+3.65	+4.38	+4.01
				$n_1=4$				
8	+1.40	+1.40	-0.72	+0.95	+0.72	-0.79	+1.09	+0.85
9	+2.00	+2.00	+0.86	+1.36	+1.29	+0.99	+1.54	+1.43
10	+2.86	+2.86	+1.22	+1.77	+1.48	+1.37	+1.98	+1.67
11	+3.75	+3.75	+1.56	+2.17	+2.09	+1.75	+2.41	+2.30
12	+4.89	+4.89	+1.89	+2.54	+2.23	+2.11	+2.83	+2.46
13	+6.10	+6.10	+2.28	+2.94	+2.80	+2.46	+3.22	+3.11
14	+7.55	+7.55	+2.70	+3.40	+3.16	+2.79	+3.60	+3.18
15	+9.08	+9.08	+3.09	+3.85	+3.68	+3.17	+4.00	+3.82
16	+10.92	+10.92	+3.47	+4.27	+4.42	+3.55	+4.42	+3.97
17	+12.86	+12.86	+3.92	+4.72	+4.54	+3.91	+4.82	+4.47
18	+15.07	+15.07	+4.36	+5.21	+6.27	+4.25	+5.19	+4.71
				$n_1=5$				
10	+2.17	+2.17	+0.99	+1.49	+1.50	+1.01	+1.56	+1.56
11	+3.14	+3.14	+1.54	+2.13	+1.83	+1.51	+2.16	+1.82
12	+4.25	+4.25	+2.13	+2.81	+2.45	+2.04	+2.77	+2.39
13	+5.67	+5.67	+2.73	+3.49	+3.10	+2.56	+3.38	+2.99
14	+7.30	+7.30	+3.33	+4.17	+3.74	+3.08	+3.98	+3.51
15	+9.27	+9.27	+3.92	+4.83	+4.36	+3.57	+4.55	+4.04
16	+11 50	+11.50	+4.47	+5.45	+5.45	+4.04	+5.08	+4.54
17	+14.15	+14.15	+5.15	+6.17	+6.57	+4.48	+5.58	+5.01
18	+17.14	+17.14	+5.86	+6.93	+7.91	+4.90	+6.05	+5.45
				$n_1=6$				
12	+3.29	+3.29	+1.79	+2.41	+2.11	+1.58	+2.24	+1.95
13	+4.50	+4.50	+2.50	+3.22	+2.95	+2.13	+2.89	+2.57
14	+6.11	+6.11	+3.25	+4.08	+3.73	+2.67	+3.54	+3.16
15	+8.00	+8.00	+4.02	+4.95	+4.53	+3.21	+4.16	+3.66
16	+10.36	+10.36	+4.77	+5.80	+5.41	+3.71	+4.75	+4.30
17	+13.08	+13.08	+5.66	+6.76	+6.85	+4.18	+5.29	+4.72
18	+16.46	+16.46	+6.63	+7.85	+8.91	+4.62	+5.80	+5.28
				$n_1=7$				
14	+4.62	+4.62	+2.97	+3.74	+3.62	+2.20	+2.98	+2.57
15	+6.33	+6.33	+3.91	+4.82	+4.47	+2.76	+3.65	+3.29
16	+8.40	+8.40	+5.00	+6.06	+5.52	+3.36	+4.35	+3.84
17	+11.00	+11.00	+6.33	+7.56	+8.06	+3.99	+5.10	+4.53
18	+14.08	+14.08	+7.73	+9.12	+9.90	+4.57	+5.79	+5.16
				$n_1=8$				
16	+6.44	+6.44	+4.71	+5.72	+6.49	+2.82	+3.73	+3.38
17	+8.60	+8.60	+6.11	+7.30	+7.15	+3.47	+4.50	+3.96
18	+11.36	+11.36	+7.80	+9.21	+10.08	+4.08	+5.24	+4.63
				$n_1=9$				
18	+8.70	+8.70	+7.48	+9.02	+10.87	+3.48	+4.54	+3.99

Table 2: No Ties

n_1+n_2	Signed(Max|Estimated-Actual|) N_0	$N_1=N_{gcd}$	E_0	$E_1=E_{gcd}$	Signed(Max(|Estimated-Actual|/Actual)) N_0	$N_1=N_{gcd}$	E_0	$E_1=E_g$
				$n_1=2$				
8	-.0516	-.0046	-.0442	-.0037	-0.363	-0.065	-0.447	-0.10
9	-.0395	-.0173	-.0348	-.0014	-0.355	-0.155	-0.390	-0.04
10	-.0375	-.0134	-.0276	-.0083	-0.341	-0.151	-0.328	-0.09
11	-.0304	-.0117	-.0250	-.0067	-0.320	+0.241	-0.303	-0.09
12	-.0377	-.0096	-.0209	-.0052	-0.293	+0.362	-0.290	+0.15
13	-.0318	-.0156	-.0258	-.0038	-0.276	+0.490	-0.274	+0.23
14	-.0268	-.0134	-.0224	-.0072	+0.295	+0.627	-0.255	+0.32
15	-.0272	-.0113	-.0208	-.0065	+0.431	+0.772	-0.233	+0.41
16	-.0235	-.0114	-.0184	-.0057	+0.573	+0.924	-0.222	+0.50
17	-.0278	-.0099	-.0211	-.0048	+0.724	+1.084	-0.216	+0.60
18	-.0246	-.0136	-.0191	-.0068	+0.881	+1.252	+0.307	+0.70
				$n_1=3$				
8	-.0351	-.0034	-.0296	-.0021	-0.293	-0.048	-0.463	-0.11
9	-.0326	-.0055	-.0204	-.0015	-0.272	+0.184	-0.405	-0.05
10	-.0224	-.0059	-.0194	-.0015	-0.251	+0.359	-0.341	+0.06
11	-.0204	-.0047	-.0168	-.0014	-0.231	+0.560	-0.271	+0.11
12	-.0218	-.0068	-.0176	+.0015	+0.381	+0.785	-0.232	+0.18
13	-.0202	-.0050	-.0157	-.0016	+0.606	+1.037	-0.217	+0.28
14	-.0198	-.0046	-.0149	+.0016	+0.856	+1.317	-0.200	+0.38
15	-.0195	-.0050	-.0124	+.0016	+1.133	+1.625	-0.187	+0.49
16	-.0158	-.0054	-.0121	+.0016	+1.437	+1.963	-0.174	+0.61
17	-.0156	-.0056	-.0116	+.0015	+1.771	+2.333	+0.258	+0.73
18	-.0152	-.0055	-.0109	+.0015	+2.136	+2.734	+0.363	+0.86
				$n_1=4$				
8	-.0255	-.0030	-.0219	+.0026	-0.271	+0.063	-0.471	-0.12
9	-.0244	-.0063	-.0215	-.0018	-0.256	+0.258	-0.421	-0.07
10	-.0179	+.0031	-.0155	+.0010	-0.229	+0.492	-0.365	+0.07
11	-.0222	-.0036	-.0180	+.0008	+0.345	+0.771	-0.306	+0.11
12	-.0200	+.0028	-.0162	+.0007	+0.628	+1.098	-0.246	+0.17
13	-.0168	-.0045	-.0136	+.0007	+0.959	+1.477	-0.204	+0.23
14	-.0153	-.0044	-.0123	+.0007	+1.341	+1.913	-0.188	+0.31
15	-.0170	-.0034	-.0130	+.0007	+1.781	+2 413	-0.169	+0.39
16	-.0149	-.0036	-.0113	+.0006	+2.285	+2.981	-0.152	+0.47
17	-.0137	-.0044	-.0104	+.0006	+2.857	+3.624	-0.140	+0.55
18	-.0124	-.0039	-.0092	+.0006	+3.504	+4.348	+0.153	+0.62
				$n_1=5$				
10	-.0238	-.0036	-.0200	+.0012	-0.223	+0.535	-0.375	+0.06
11	-.0167	-.0033	-.0142	+.0009	+0.425	+0.874	-0.331	+0.10
12	-.0173	-.0041	-.0142	+.0007	+0.775	+1.283	-0.293	+0.15
13	-.0173	-.0033	-.0137	+.0006	+1.198	+1.774	-0.264	+0.20
14	-.0147	-.0037	-.0115	+.0004	+1.703	+2.357	-0.249	+0.24
15	-.0143	-.0040	-.0112	+.0004	+2.303	+3.046	-0.254	+0.28
16	-.0143	-.0032	-.0109	+.0004	+3.012	+3.854	-0.286	+0.31
17	-.0120	-.0035	-.0093	+.0004	+3.844	+4.797	-0.350	+0.32
18	-.0121	-.0029	-.0910	+.0003	+4.814	+5.892	-0.454	+0.32
				$n_1=6$				
12	-.0151	-.0031	-.0126	+.0007	+0.824	+1.345	-0.311	+0.14
13	-.0179	-.0030	-.0114	+.0006	+1.317	+1.922	-0.313	+0.17
14	-.0159	-.0030	-.0105	+.0005	+1.922	+2.626	-0.346	+0.19
15	-.0145	-.0028	-.0096	+.0004	+2.661	+3.479	-0.423	+0.19
16	-.0134	-.0027	-.0103	+.0003	+3.557	+4.508	-0.559	+0.16
17	-.0124	-.0027	-.0094	+.0003	+4.639	+5.742	-0.769	-0.28
18	-.0116	-.0026	-.0088	+.0003	+5.935	+7.212	-1.000	-0.57
				$n_1=7$				
14	-.0145	-.0025	-.0116	+.0004	+1.995	+2.715	-0.382	+0.17
15	-.0122	-.0030	-.0099	+.0003	+2.841	+3.698	-0.520	+0.13
16	-.0124	-.0025	-.0097	+.0003	+3.892	+4.909	-0.753	-0.30
17	-.0107	-.0028	-.0084	+.0003	+5.188	+6.394	-1.000	-0.65
18	-.0109	-.0024	-.0083	+.0002	+6.779	+8.205	-1.000	-1.00
				$n_1=8$				
16	-.0114	-.0028	-.0090	+.0003	+4.004	+5.044	-0.822	-0.37
17	-.0108	-.0028	-.0085	+.0002	+5.468	+6.726	-1.000	-0.85
18	-.0102	-.0027	-.0079	+.0002	+7.300	+8.818	-1.000	-1.00
				$n_1=9$				
18	-.0096	-.0025	-.0075	+.0002	+7.476	+9.025	-1.000	-1.00

A COMPARISON OF SPEARMAN'S FOOTRULE AND RANK CORRELATION COEFFICIENT WITH EXACT TABLES AND APPROXIMATIONS

LeRoy A. Franklin, Indiana State University

ABSTRACT

Spearman's Footrule, D, is the sum of the absolute values of the differences between the ranks in two rankings of n objects. For the case of equally likely permutations, tables of the exact cumulative distribution function (c.d.f.) of D are given for $11 \le n \le 18$. For both Spearman's Footrule and Rank Correlation Coefficient the maximum difference between the exact c.d.f. and the normal approximation is given as well as the maximum difference between the exact c.d.f. and the normal approximation with correction for continuity and comparisons made.

1. Introduction

Given two rankings of n objects or equivalently, two permutations p and q, a widely used non-parametric measure of association between the rankings is Spearman's S given in unnormalized form as S, where

$$S(p,q) = \sum_{i=1}^{n} (p_i - q_i)^2. \qquad (1.1)$$

S is related to Spearman's Rank Correlation Coefficient by $r_s = 1 - 6S/(n^3-n)$ and its derivation along with many of its properties and moments can be found in Kendall (1970). In particular it is shown there that S is distributed from 0 to $(n^3-n)/3$ on the even integers with a mean of $(n^3-n)/6$ and variance of $(n^3-n)^2/36(n-1)$.

An equally simple but neglected competitor is Spearman's (1904) footrule given by D, where

$$D(p,q) = \sum_{i=1}^{n} |p_i - q_i|. \qquad (1.2)$$

The footrule has historically not been an important measure of association because of a lack of desirable statistical properties (cf. Pearson, 1907 and Kendall, 1970). Diaconis and Graham (1977) recently revived interest in D by treating it as a metric on the set of permutations, establishing its limiting normality by use of Hoeffding's (1951) combinatorial central limit theorem and show it to be related to Kendall's τ given by T, where

$$T(p,q) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{sign}(p_i - p_j)\,\text{sign}(q_i - q_j). \qquad (1.3)$$

They were able to show $T \le D \le 2T$ and concluded that S is probably the better metric with D and T roughly the same. While D is somewhat easier to interpret directly, T had the advantage of having the exact table tabulated for the c.d.f. for small sample sizes (cf. Kendall, 1970).

Ury and Kleinecke (1979) tabulated the exact c.d.f. for D for n = 2(1)10 and gave an approximate table for the c.d.f. for D for n=11(1)15 generated by Monte Carlo approximation. They also conjectured about the rate of convergence to an approximating normal distribution and that an improvement in the approximation could be accomplished by using a standard half-interval continuity correction of +1 to be applied for the c.d.f. of D. Diaconis and Graham (1977) calculated the asymptotic mean and variance of D and showed $0 \le D \le n^2/2$ for n even and $0 \le D \le (n^2-1)/2$ for n odd on the even integers. Spearman (1904) and Kleinecke, Ury and Wagner (1962) derived the exact mean and variance of D given by

$$E(D) = (n^2 - 1)/3 \qquad (1.4)$$

and

$$\text{Var}(D) = (n+1)(2n^2 + 7)/45. \qquad (1.5)$$

2. Calculation of Exact Tables of the Null Distribution of S and D for n = 11(1)18

Assuming the null distribution of S or D means all possible n! permutations are equally likely. By calculating each possible permutation and comparing it to a base ranking 1,2,3 .., n the corresponding S or D can be calculated and counted, yielding the frequency distribution of either. Dividing each frequency by n! gives the corresponding probability density function for S or D and summing yields the exact c.d.f. While conceptually easy, a straight forward approach for calculating the exact c.d.f. when n = 17 requires 35, 568, 728, 096, 000 permutations to be calculated, each involving the finding the sum of 17 terms of absolute values of differences for D or the sum of 17 terms of squared differences for S. This would involve staggering amounts of computer time. However, Table 1, which displays the exact c.d.f. for D for n up to 18, was calculated utilizing permutations, combinations, and stored arrays as outlined by Franklin (1987). In that paper the exact c.d.f. for S was presented for n = 12(1)16 using the same method. A later paper by Franklin (1987) extended the table for S to n = 17 and 18 again using this technique.

In this technique first k was chosen (approximately n/2) and then all the k! and (n-k)! permutations of {1,2,..,k} and {1,2,..,n-k} were stored in matrices A and B (respectively). Next, a k-sized combination of integers from {1,2,...n} and its resulting combination of (n-k) integers (the "remainder") was determined. All possible k! permutations were formed of the k-sized combination with each permutation compared to the base ranking of 1,2,..,k and a corresponding sum of absolute differences calculated and stored in a matrix S_1. Then all possible (n-k)! permutations of the "remainder" were found with each permutation compared to the base ranking of k+1,k+2,..,n and a corresponding

sum of absolute differences calculated and stored in a matrix S_2. Multiplying the counts of $S_1(i)$ by the counts of $S_2(j)$ and placing the resulting count in $S(i+j)$ results in an equivalent process of $(k!) \times (n-k)!$ permutations for S. Finally summation over all $\binom{n}{k}$ combinations gives the complete n! permutations for the null distribution. (Copies of the program are available from the author.) All calculations were done in quadruple precision allowing accuracy of 1×10^{-14} on a Harris 1000 "Super Mini" computer. Internal checks on the number of permutations and external, theoretical checks on the cumulative distribution assured complete accuracy. Use of this technique allows several orders of magnitude of decrease in calculation time.

Comparison of the exact distribution of Table 1 with the table produced by Monte Carlo simulation by Ury and Kleineche (1979) shows remarkable accuracy. The largest difference was .003 with most entries differing by .001 or less for $n \leq 15$.

3. Normal Approximations to the Exact Distribution of S and D

In order to determine the precise degree of convergence in distribution of S and D to normality, two approximations to the exact c.d.f. of both were investigated by numerical integration: the normal approximation and the normal approximation with continuity correction (c.c.). Table 2 displays the maximum (c.d.f. exact − c.d.f. approx.) over all possible values of S and D for both approximations for n = 9 through 18.

Ury and Kleinecke (1979) stated that "despite the asymmetry of D, the tendency toward the normal distribution is rather fast (much faster than R)," Spearman's Rho. And that "the approximation is quite good for n = 10." However, examination of Table 2 show the contrary. For the uncorrected normal approximation for n = 10, the error can be as large as .046 and is over .025 for $22 \leq D \leq 44$. These values of D occur with probability over .93. Even for n=18, the uncorrected normal approximation can have a maximum absolute error as large as .019. However, substantial improvement is obtained by using the normal approximation with c.c. with maximum absolute error of .009 for n = 18.

Furthermore, while for D and S the maximum absolute error decreases monotonically as n increases for both approximations, the convergence of D is rather slow compared to S. Comparing the same maximum absolute error as presented by Table 2 for corresponding n shows the normal approximation of S to have less than half the error of the normal approximation of D (.046 for D versus .018 for S at n = 10 and .019 for D versus .008 for S at n = 18). Comparing the maximum absolute error for corresponding n when the normal approximations have the continuity correction factor shows S to have about 2/3 the error of D (.015 for D versus .012 for S at n = 10 and .009 for D versus .006 for S when n = 18). The extra error of D seems to be attributable to the asymmetry of distribution of the footrule that seems significantly present for even n = 18.

4. Conclusion and Recommendations

The exact c.d.f. of Spearman's footrule and Spearman's Rank Correlation coefficient should be used for $n \leq 18$, since they now exist. For $n \geq 19$, the straight normal approximation for D should be avoided in favor of the normal approximation with continuity correction factor. For all $n \geq 19$, such an approximation will have an error of $< .006$ for any value of D and will have even smaller error ($\leq .003$) for most upper and lower tail values. Convergence to normality of Spearman's footrule is significantly slower than the convergence of Spearman's Rho.

Seven different approximations to S have been presented by Franklin (1987). The clearly and dramatically superior approximation was shown in that paper to be a Pearson Type II approximation. For further discussion of that approximation see Olds (1938) and Zar (1972). For that approximation the maximum absolute error is given by .000158 for n = 18.

REFERENCES

Diaconis, P. and Graham, R. L. (1977). Spearman's Footrule as a Measure of Disarray. J. R. Statist. Soc. B, 39, 262-268.

Franklin, L. A. (1987). The Complete Exact Null Distribution of Spearman's Rho for n = 12(1) 16. Proceedings of the 19th Symposium on the Interface Between Statistics and Computer Science, March 1987.

Franklin, L. A. (1987). Approximations, Convergence and Exact Tables for Spearman's Rank Correlation Coefficient. Proceedings of the Statistical Computation Section, American Statistical Association, Fall, 1987.

Hoeffding, W. (1951). A Combinatorial Central Limit Theorem. Ann. Math. Statist., 22, 558 566.

Kendall, M. G. (1970). Rank Correlation Methods, 4th ed. London: Griffin.

Kleinecke, D. C., Ury, H. K., and Wagner, L. F. (1962). Spearman's Footrule--an Alternative Rank Statistic. Civil Defense Research Project, Institute of Engineering Research, University of California-Berkeley, Report No. CDRP-182-114, November 1962.

Olds, E. G. (1938). Distribution of Sums of Squares of Rank Difference for Small Numbers of Individuals. Annals of Mathematical Statistics, 9, 133-148.

Pearson, K. (1907). Mathematical Contributions to the Theory of Evolution. XVI. On Further Methods of Determining Correlation. Draper's Co. Res. Mem., Biometric Series IV. Cambridge University Press.

Spearman, C. (1904). The Proof and Measurement of Association Between Two Things. Amer. J. Psychol., 15, 72-101.

_____ (1906). "Footrule" for Measuring Correlation. Brit. J. Psychol., 2, 89-108.

Ury, H. K. and Kleinecke, D. C. (1979). Tables of the Distribution of Spearman's Footrule. Applied Statistics, 28, 271 275.

Zar, Jerroes H. (1972). Significance Testing of the Spearman's Rank Correlation Coefficient. Journal of the American Statistical Association, 67, 578 580.

Table 1

Exact cumulative null distributor of D

d/n	11	12	13	14	15	16	17	18
0	0.25-7	0.21-8	0.16-9	0.1-10	0.8-12	0.5-13	0.3-14	0.2-15
2	0.28-6	0.25-7	0.21-8	0.16-9	0.1-10	0.8-12	0.5-13	0.3-14
4	0.19-5	0.18-6	0.16-7	0.13-8	0.10-9	0.7-11	0.5-12	0.3-13
6	0.93-5	0.98-6	0.93-7	0.81-8	0.65-9	0.5-10	0.3-11	0.2-12
8	0.38-4	0.43-5	0.44-6	0.40-7	0.34-8	0.27-9	0.2-10	0.1-11
10	0.13-3	0.16-4	0.17-5	0.17-6	0.15-7	0.13-8	0.10-9	0.7-11
12	0.39-3	0.52-4	0.61-5	0.63-6	0.60-7	0.53-8	0.42-9	0.3-10
14	0.0011	0.15-3	0.19-4	0.21-5	0.21-6	0.19-7	0.16-8	0.13-9
16	0.0026	0.39-3	0.53-4	0.63-5	0.67-6	0.65-7	0.58-8	0.47-9
18	0.0056	0.94-3	0.14-3	0.17-4	0.19-5	0.20-6	0.19-7	0.16-8
20	0.0113	0.0021	0.32-3	0.44-4	0.52-5	0.57-6	0.55-7	0.50-8
22	0.0211	0.0042	0.70-3	0.10-3	0.13-4	0.15-5	0.15-6	0.14-7
24	0.0368	0.0080	0.0014	0.23-3	0.31-4	0.37-5	0.40-6	0.40-7
26	0.0606	0.0143	0.0028	0.47-3	0.67-4	0.86-5	0.99-6	0.10-6
28	0.0946	0.0244	0.0051	0.92-3	0.14-3	0.19-4	0.23-5	0.25-6
30	0.1403	0.0395	0.0090	0.0017	0.28-3	0.40-4	0.51-5	0.58-6
32	0.1990	0.0611	0.0150	0.0030	0.53-3	0.80-4	0.11-4	0.13-5
34	0.2700	0.0907	0.0240	0.0052	0.96-3	0.15-3	0.22-4	0.27-5
36	0.3522	0.1295	0.0370	0.0086	0.0017	0.28-3	0.42-4	0.55-5
38	0.4420	0.1782	0.0550	0.0137	0.0028	0.51-3	0.79-4	0.11-4
40	0.5363	0.2369	0.0791	0.0210	0.0046	0.87-3	0.14-3	0.20-4
42	0.6295	0.3049	0.1102	0.0314	0.0073	0.0015	0.25-3	0.38-4
44	0.7180	0.3807	0.1490	0.0454	0.0113	0.0024	0.42-3	0.67-4
46	0.7955	0.4619	0.1958	0.0640	0.0169	0.0037	0.70-3	0.12-3
48	0.8617	0.5455	0.2506	0.0878	0.0246	0.0057	0.0011	0.20-3
50	0.9120	0.6281	0.3127	0.1174	0.0350	0.0086	0.0018	0.32-3
52	0.9498	0.7063	0.3808	0.1535	0.0486	0.0126	0.0027	0.52-3
54	0.9740	0.7771	0.4531	0.1960	0.0660	0.0180	0.0041	0.81-3
56	0.9895	0.8383	0.5277	0.2451	0.0877	0.0253	0.0061	0.0013
58	0.9960	0.8888	0.6018	0.3001	0.1144	0.0348	0.0088	0.0019
60	1.0000	0.9281	0.6735	0.3602	0.1462	0.0470	0.0125	0.0028
62		0.9569	0.7400	0.4243	0.1834	0.0623	0.0173	0.0041
64		0.9766	0.7999	0.4908	0.2259	0.0812	0.0237	0.0058
66		0.9886	0.8515	0.5580	0.2736	0.1040	0.0319	0.0082
68		0.9953	0.8946	0.6242	0.3258	0.1310	0.0422	0.0113
70		0.9989	0.9284	0.6876	0.3819	0.1624	0.0550	0.0154
72		1.0000	0.9545	0.7466	0.4408	0.1984	0.0706	0.0206
74			0.9725	0.7999	0.5013	0.2387	0.0894	0.0273
76			0.9850	0.8467	0.5623	0.2832	0.1115	0.0356
78			0.9925	0.8863	0.6222	0.3314	0.1373	0.0458
80			0.9971	0.9188	0.6799	0.3828	0.1168	0.0583
82			0.9989	0.9443	0.7340	0.4364	0.2001	0.0732
84			1.0000	0.9636	0.7838	0.4915	0.2371	0.0907
86				0.9776	0.8282	0.5471	0.2777	0.1112
88				0.9871	0.8670	0.6022	0.3214	0.1348
90				0.9932	0.8998	0.6556	0.3678	0.1615
92				0.9968	0.9269	0.7066	0.4164	0.1914
94				0.9987	0.9484	0.7542	0.4665	0.2246
96				0.9997	0.9650	0.7978	0.5174	0.2607
98				1.0000	0.9772	0.8369	0.5682	0.2997
100					0.9861	0.8712	0.6182	0.3413
102					0.9919	0.9006	0.6667	0.3849
104					0.9957	0.9252	0.7129	0.4302
106					0.9979	0.9452	0.7562	0.4766
108					0.9992	0.9611	0.7961	0.5234
110					0.9997	0.9733	0.8322	0.5702
112					1.0000	0.9824	0.8643	0.6163
114						0.9889	0.8923	0.6611
116						0.9934	0.9162	0.7040
118						0.9963	0.9362	0.7445
120						0.9981	0.9526	0.7822
122						0.9991	0.9656	0.8168
124						0.9996	0.9758	0.8481

Table 1 - Continued
Exact cumulative null distributor of D

d/n	11	12	13	14	15	16	17	18
126						0.9999	0.9835	0.8759
128						1.0000	0.9892	0.9002
130							0.9932	0.9211
132							0.9959	0.9388
134							0.9977	0.9535
136							0.9988	0.9654
138							0.9994	0.9748
140							0.9997	0.9821
142							0.9999	0.9877
144							1.0000	0.9918
146								0.9947
148								0.9968
150								0.9981
152								0.9990
154								0.9995
156								0.9997
158								0.9999
160								0.9999
162								1.0000

Table 2
Maximum | c.d.f. exact - c.d.f. approx. | Over All Possible Values of D

n	10	11	12	13	14	15	16	17	18
Normal Approximation at D =	.046 30	.040 36	.035 42	.031 50	.028 58	.025 68	.023 76	.021 86	.019 96
Normal Approximation with C.C. at D =	.015 34	.014 42	.013 50	.012 58	.011 68	.011 78	.010 88	.010 100	.009 112

THE EFFECTS OF HEAVY TAILED DISTRIBUTIONS ON THE TWO-SIDED k-SAMPLE SMIRNOV TEST

Henry D. Crockett and M. M. Whiteside, University of Texas at Arlington

Abstract: This paper presents the problem that the k-sample Smirnov test has in discriminating the ranking of samples from heavy tailed probability distribution functions. The test results for 1000 tests are presented for each of seven levels of variance and five scaler offsets for the two distributions.

1. **Introduction**. Given nk random variables $\{X_{ij}\}$, i=1,..,n,j=1,..,k, which represent k random samples of equal size n from an absolutely continuous distribution function $F_i(x)$. The k-sample Smirnov test is used to determine if the population distribution functions F(x) are identical. Thus the hypotheses would be:

$$H_0 : F_1(x) = F_2(x) = \ldots = F_k(x) \text{ for all } x,$$

$$H_1 : F_t(x) = F_u(x) \text{ for some t, u, and x.}$$

In order to perform the test, the sample must be ordered within themselves, so that the rth ordered sample is $Z_{1r} < Z_{2r},..<Z_{nr}$. Then Z_{ir} is the ith order statistic from the rth sample, and Z_{nr} is the extreme of the rth sample. Thus the empirical distribution function of the rth sample would be:

$$F_r(x) = 0 \quad \text{if } x < Z_{ir},$$

$$F_r(x) = a/n \quad \text{if } Z_{ar} <= x <= A_{a+1,r},$$

$$F_r(x) = 1 \quad \text{if } Z_{nr} <= x.$$

The samples are then ordered among themselves on the basis of their most extreme points. Therefore, if S is the set of extremes from the k-samples such that $S = \{Z_{nr}, r = 1,..,k\}$. The set S is then ordered to determine the smallest and the largest Z_{nr}, the sample related to the smallest Z_{nr} is then the sample of rank 1 or $F^{(1)}(x)$, and the sample related to the largest Z_{nr} is the sample of rank k or $F^{(k)}(x)$. The test statistic T_1 is then defined by Conover (1980) as the maximum vertical distance between $F^{(1)}(x)$ and $F^{(k)}(x)$. Mathematically this is stated as:

$$T_1 = \sup_x [F^{(1)}(x) - F^{(k)}(x)].$$

This test statistic would then be compared to a table value to obtain a decision for a given level of α.

2. **Effect of Heavy Tailed Distributions**. The method for choosing the largest and smallest sample for comparison appears to be susceptible to error when choosing among heavy tailed distributions. These distributions are more likely to have extreme values due to the nature of their probability distribution functions (p.d.f.s) than a more leptokurtic p.d.f. This would indicate that if k samples were drawn from populations with the same p.d.f.s, of which one differed from the others only by some scaler factor, the p.d.f.s with heavier tailed distributions would choose the true "largest" sample less frequently than the p.d.f.s with smaller tails. In this way, extreme values in the samples could greatly affect which one of the samples was chosen as the "largest" sample (Fig. 1). Therefore, since the true "largest" sample is determined less often, the test statistic is comparing two samples from truly equivalent distributions; therefore, the probability of failing to reject should equal $1-\alpha$, when in fact the null hypothesis is false.

In order to show how these differences affect the two-sided K-sample Smirnov test, a simulation was performed. The two p.d.f.s which were compared were the uniform distribution and the double exponential distribution. These were performed on sample sizes of 6, 12, and 30. For each distribution and each sample size, several levels of variance and scaler factor movement of one of the three samples were considered. The levels of variance were 10, 25, 50, 75, 100, 150, and 200. The scaler factor added to one of the samples drawn were 0, 1, 2, 4, and 8. Random number generation and test results for 1000 tests for each level of sample size, variance, and factor were performed using the Statistical Applications System (SAS). The null hypothesis is that the population distribution functions are identical at the α=0.05 level of significance.

3. **Simulation Results**. The result of the simulation were mixed. The expected results would be that the Smirnov test would detect a scaler factor change in one sample more often for a uniform distribution, and would therefore reject the null hypothesis more often than the samples from double exponentials. These tests were all performed at the α=0.05 level and the percentage of rejections were determined for each level (Tables 1-6). Also for each level the percentage of times the sample which had a scaler factor change was chosen as the largest sample is shown in parentheses. The results appear to be conflicting because although for each level of all factors the uniform distribution correctly chose the true largest sample more frequently than the double exponential, it also appears that the rejection rate was larger for the double exponential than for uniform distributions.

Upon closer inspection of the distribution functions, however, this result appears to be justified. From Figures 2 and 3, it is apparent that given two distributions functions of uniform density, one of which has been adjusted by a scaler, the distance between the two distributions is much less than two double exponentials that have been separated by the same amount. Therefore, even though the correct largest sample was chosen more often from the samples of the univarite p.d.f.s, the decision to reject the null hypothesis may not have been made due to smaller values of T_1.

4. **Conclusions and Speculation**. Since the assumption that keptokurtic distributions will correctly identify a similar distribution function which has been adjusted by an offset

more often than a platykurtic distribution was shown to be correct, further investigation is warranted. The next step in investigations will be to compare distributions which are more similar in shape, but one of which has more extreme values than the other. This will probably show that distributions which have more extreme values also have a much greater chance for Type II errors than those without. Another avenue of investigation might be appropriate in order to determine a methodology of choosing the largest and smallest sample which is less susceptible to extreme values. A suggestion for this might be to obtain the three largest observations from each sample, rank these, and then determine which samples have the smallest and largest sum of ranks. This would appear to address the problem of an outlier in one of the sets of samples.

References

Birnbaum, Z. W., and R. A. Halls, "Small Sample Distributions for Multi-Sample Statistics of the Smirnov Type," Annals of Mathematic Statistics, Vol. 31, pp. 710-720.

Conover, W. J., Practical Nonparametric Statistics (2nd ed.), John Wiley & Sons, 1980, pp. 382-384.

Conover, W. J.., "Several k-Sample Kolmogorov-Smirnov Tests," Annals of Mathematic Statistics, Vol. 36, pp. 1019-1026.

David, Herbert T., "A Three-Sample Kolmogorov-Smirnov Test," Annals of Mathematic Statistics, Vol. 29, pp. 842-851.

Kiefer, J., "K-Sample Analogues of the Kolmogorov-Smirnov and Cramer-V. Mises Tests," Annals of Mathematic Statistics, Vol. 30, pp. 420-447.

Figure 1

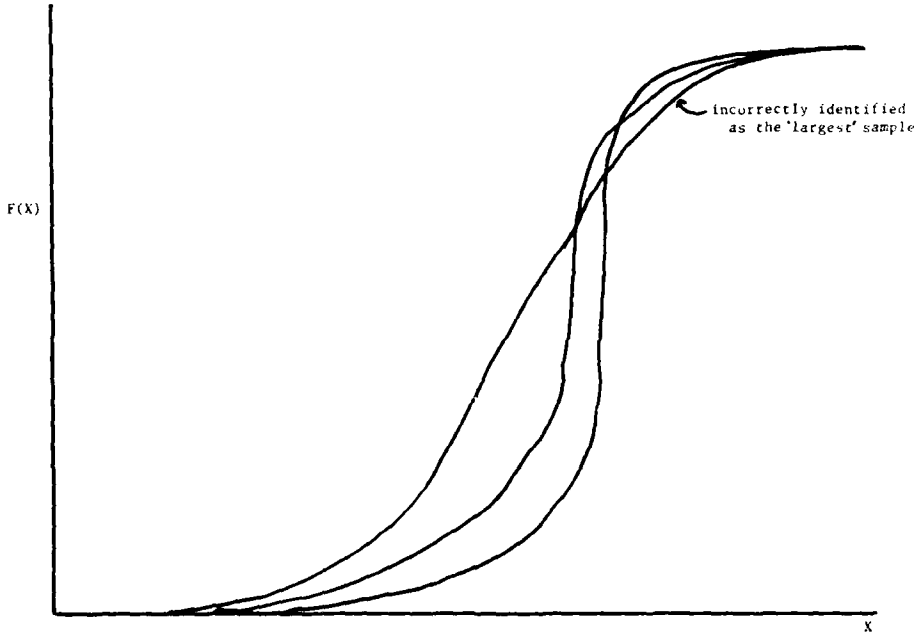


Table 1

UNIFORM DISTRIBUTION (k=3,n=6)

V A R I A N C E	FACTOR 0	1	2	4	8
10	0.044 (0.347)	0.055 (0.646)	0.104 (0.826)	0.323 (0.962)	0.951 (1.000)
25	0.038 (0.331)	0.047 (0.562)	0.066 (0.688)	0.128 (0.883)	0.468 (0.996)
50	0.036 (0.325)	0.043 (0.489)	0.063 (0.598)	0.085 (0.779)	0.268 (0.950)
75	0.035 (0.342)	0.039 (0.453)	0.031 (0.581)	0.072 (0.724)	0.205 (0.929)
100	0.041 (0.334)	0.036 (0.431)	0.036 (0.538)	0.058 (0.679)	0.151 (0.874)
150	0.044 (0.308)	0.042 (0.457)	0.058 (0.485)	0.063 (0.659)	0.130 (0.809)
200	0.034 (0.332)	0.041 (0.417)	0.039 (0.487)	0.048 (0.612)	0.080 (0.788)

Table 2

DOUBLE EXPONENTIAL (k=3,n=6)

V A R I A N C E	FACTOR 0	1	2	4	8
10	0.040 (0.357)	0.067 (0.429)	0.168 (0.567)	0.536 (0.784)	0.932 (0.957)
25	0.036 (0.336)	0.051 (0.390)	0.085 (0.459)	0.238 (0.619)	0.637 (0.814)
50	0.044 (0.340)	0.047 (0.376)	0.069 (0.432)	0.144 (0.512)	0.428 (0.727)
75	0.039 (0.346)	0.042 (0.367)	0.058 (0.409)	0.114 (0.497)	0.303 (0.644)
100	0.040 (0.351)	0.046 (0.336)	0.045 (0.373)	0.090 (0.486)	0.238 (0.644)
150	0.043 (0.333)	0.049 (0.338)	0.038 (0.377)	0.062 (0.444)	0.170 (0.580)
200	0.037 (0.334)	0.036 (0.341)	0.049 (0.371)	0.069 (0.421)	0.137 (0.543)

Table 3

UNIFORM DISTRIBUTION (k=3,n=12)

VARIANCE \ FACTOR	0	1	2	4	8
10	0.032 (0.346)	0.070 (0.783)	0.171 (0.934)	0.638 (0.999)	1.000 (1.000)
25	0.031 (0.312)	0.057 (0.649)	0.099 (0.854)	0.273 (0.972)	0.833 (1.000)
50	0.029 (0.333)	0.038 (0.603)	0.065 (0.761)	0.158 (0.926)	0.517 (0.993)
75	0.032 (0.346)	0.044 (0.560)	0.060 (0.699)	0.111 (0.911)	0.339 (0.989)
100	0.033 (0.350)	0.037 (0.538)	0.062 (0.697)	0.081 (0.838)	0.244 (0.973)
150	0.031 (0.312)	0.044 (0.482)	0.045 (0.644)	0.058 (0.813)	0.195 (0.950)
200	0.029 (0.333)	0.033 (0.471)	0.048 (0.594)	0.074 (0.765)	0.162 (0.913)

Table 4

DOUBLE EXPONENTIAL (k=3,n=12)

VARIANCE \ FACTOR	0	1	2	4	8
10	0.03 (0.328)	0.102 (0.456)	0.283 (0.562)	0.696 (0.778)	0.935 (0.931)
25	0.035 (0.341)	0.069 (0.388)	0.112 (0.464)	0.413 (0.628)	0.809 (0.829)
50	0.037 (0.340)	0.032 (0.363)	0.079 (0.436)	0.214 (0.518)	0.612 (0.726)
75	0.030 (0.328)	0.043 (0.382)	0.056 (0.412)	0.155 (0.495)	0.477 (0.638)
100	0.024 (0.328)	0.032 (0.368)	0.054 (0.400)	0.117 (0.475)	0.418 (0.612)
150	0.035 (0.341)	0.048 (0.341)	0.043 (0.368)	0.095 (0.436)	0.281 (0.535)
200	0.037 (0.340)	0.025 (0.339)	0.041 (0.374)	0.080 (0.406)	0.208 (0.530)

Table 5

UNIFORM DISTRIBUTION (k=3,n=30)

VARIANCE \ FACTOR	0	1	2	4	8
10	0.026 (0.337)	0.109 (0.978)	0.388 (0.998)	0.969 (1.000)	1.000 (1.000)
25	0.026 (0.337)	0.058 (0.885)	0.166 (0.982)	0.639 (0.999)	0.998 (1.000)
50	0.038 (0.325)	0.046 (0.821)	0.104 (0.945)	0.323 (0.997)	0.924 (1.000)
75	0.026 (0.337)	0.030 (0.771)	0.068 (0.908)	0.231 (0.991)	0.748 (1.000)
100	0.038 (0.325)	0.039 (0.735)	0.062 (0.884)	0.171 (0.983)	0.619 (1.000)
150	0.026 (0.337)	0.025 (0.696)	0.057 (0.820)	0.136 (0.966)	0.412 (0.999)
200	0.038 (0.325)	0.036 (0.638)	0.050 (0.820)	0.092 (0.941)	0.315 (0.999)

Table 6

DOUBLE EXPONENTIAL (k=3,n=30)

VARIANCE \ FACTOR	0	1	2	4	8
10	0.025 (0.335)	0.146 (0.426)	0.460 (0.546)	0.742 (0.739)	0.951 (0.950)
25	0.033 (0.300)	0.066 (0.376)	0.244 (0.484)	0.592 (0.609)	0.828 (0.825)
50	0.025 (0.335)	0.050 (0.365)	0.132 (0.429)	0.388 (0.502)	0.737 (0.733)
75	0.033 (0.300)	0.035 (0.343)	0.097 (0.432)	0.294 (0.474)	0.629 (0.641)
100	0.025 (0.335)	0.037 (0.354)	0.074 (0.400)	0.228 (0.453)	0.610 (0.632)
150	0.033 (0.300)	0.030 (0.337)	0.056 (0.405)	0.172 (0.427)	0.464 (0.547)
200	0.025 (0.335)	0.034 (0.342)	0.043 (0.382)	0.132 (0.408)	0.416 (0.543)

Figure 3

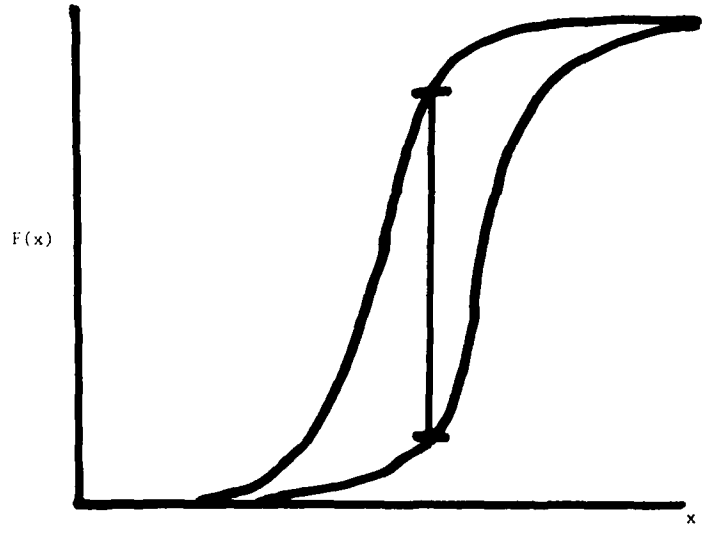
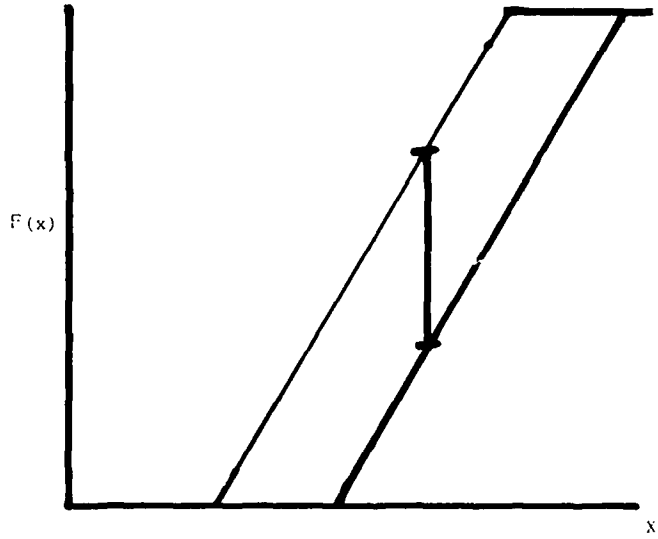


Figure 2



SIMULATED POWER COMPARISONS OF MRPP RANK TESTS AND SOME STANDARD SCORE TESTS

Derrick S. Tracy and Khushnood A. Khan, University of Windsor

Abstract

Two MRPP rank tests and two standard score tests - median and Fisher, are compared with respect to their empirical powers, computed from extensive simulations from normal, Cauchy and Laplace underlying populations. This is done for several combinations of sample sizes - unequal and equal.

In applying classical linear rank tests to meteorological data, Mielke, Berry and Medina (1982) posed the problem that in most cases the analysis space associated with these tests is non-metric, and hence the p-values may not be interpreted correctly. An alternative inference technique known as multiresponse permutation procedure (MRPP) is proposed, of which a generalized version is discussed in Mielke (1984).

Let $\Omega = \{w_1,\ldots,w_N\}$ be a finite population of N objects, each of which has r responses, and the responses have the same range via a rank order transformation. We let $K = \sum_1^g N_i$ of these be classified into g mutually exclusive subgroups according to some *a priori* classification scheme. The excess N-K observations are in the $(g+1)^{th}$ subgroup. Then the MRPP test statistic is defined as $\delta = \sum_1^g C_i\xi_i$, where $C_i>0$, $\sum_1^g C_i=1$, $\xi_i = \binom{N_i}{2}^{-1} \sum_{I<J}^{N} \Delta_{IJ}\, S_i(w_I)S_i(w_J)$, $S_i(w_I)$ being an indicator function, and Δ_{IJ} the symmetric distance function $\left(\sum_{k=1}^{r} |w_{kI}-w_{kJ}|^p\right)^{\nu/p}$, $p\geq 1$, $\nu>0$. (When r=1, p is irrelevant.) The analysis space is non-metric for $\nu>1$ and metric for $\nu\leq 1$. The majority of the permutation tests used in practice are based on $\nu=2$. The choice of p=2 and $\nu=1$ is recommended.

Under H_o: classification is random, equal probabilities are assigned to the $M = N!/\prod_1^{g+1}(N_i!)$ possible allocations of the N objects into the subgroups. H_o is rejected when δ is small. The number M of possible allocations is very large, even for moderate N, making it difficult to obtain the exact distribution. This is overcome by taking the approximate distribution, using the first four moments of δ, see Tracy and Tajuddin (1985). An efficient choice of C_i when the N_i's are not equal is N_i/K, as suggested by Mielke (1984, p.817). Mielke, Berry, Brockwell and Williams (1981) considered a special case of Δ_{IJ} when r=1 and measurements are replaced by their ranks $R(w_I)$ in the combined sample. Then $\Delta_{IJ} = |R(w_I)-R(w_J)|^{\nu}$, $\nu>0$. For $\nu=1,2$, they denoted the test statistic by δ_1, δ_2. Using the Pearson criterion, appropriate Pearson Type curves are suggested by Tracy and Tajuddin (1985), and power performance studied for equal sample sizes by Tracy and Tajuddin (1986). For $\nu\neq 2$, Brockwell, Mielke and Robinson (1982) show that δ has a non-normal non-invariant distribution, and that its asymptotic distribution depends on the underlying population.

In this paper, we compare the power performance of δ_1 and δ_2 for N_1, N_2 unequal and equal with N=80 when the underlying distributions are normal, Cauchy and Laplace. Based on a simulation study, 5000 samples are generated using IMSL subroutines. The first N_1 observations are shifted by $k\sigma$, where k proceeds from 0 to 1.0, at steps of 0.2. To obtain empirical powers, the number of rejections is counted for α = .001, .01, .05 and .10. The appropriate approximations for the distributions of δ_1 and δ_2 are Pearson Type VI and Type I respectively. The empirical powers of two standard score tests - median and Fisher, are also studied for comparison purposes. The results are presented in Tables 1 - 3, and some typical graphs are also drawn. On interchanging the roles of N_1 and N_2, the powers of the test statistics remain more or less the same for symmetric underlying populations. Hence we only present the case of $N_1\geq N_2$, for N_1=70,60,50,40.

If we were to use only the first three moments of δ_1, δ_2, the appropriate approximation to their distribution is Pearson Type III, see Mielke, Berry, Brockwell and Williams (1981). For α = .001, the powers of δ_1 under Pearson Type III approximation are higher than those under Type VI approximation for all shifts and sample sizes. However, for α = .01, .05 and .10, the powers of δ_1 under Type VI approximation are slightly higher than those under Type III approximation for all sample sizes except (N_1,N_2)=(70,10), (10,70). Powers of δ_2 under Type I approximation are slightly lower than those under Type III approximation for α = .01, .05 and .10, but the situation is the other way round for α = .001. For $N_1 = N_2$ and α = .01, both approximations give the same value.

For normal underlying populations, powers of δ_2 are higher than those for δ_1. For α = .001, the power of Fisher test is lower than that of δ_2 but generally higher than that of δ_1. For α = .01 and for lower shifts the power of the Fisher test is lower, but for higher shifts and for all other α's, the powers of Fisher test are higher.

For heavy-tailed distributions (Cauchy and Laplace), the powers of δ_1 are higher than those of δ_2. Powers of Fisher test are lower than those of δ_1, δ_2 and median test.

In most cases the empirical levels of significance for the median test are not within 2 standard deviation limits, and hence should not be compared. However, where comparable, the median test has slightly high power than the other tests, but for underlying normal population it gains the least powers.

Table 1. Empirical Powers when the Underlying Distribution is NORMAL

N_1:	70				60				50				40			
N_2:	10				20				30				40			
Shift	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher
							$\alpha = 0.001$									
0.0	.0008	.0008	.0008	.0006	.0012	.0012	.0004	.0010	.0008	.0008	.0002	.0006	.0004	.0006	.0002	.0004
0.2	.0032	.0040	.0030	.0026	.0066	.0070	.0026	.0064	.0076	.0074	.0022	.0064	.0072	.0074	.0036	.0072
0.4	.0130	.0140	.0066	.0124	.0282	.0322	.0102	.0274	.0468	.0506	.0152	.0438	.0182	.0550	.0230	.0472
0.6	.0446	.0482	.0232	.0408	.1256	.1342	.0480	.1220	.1890	.2006	.0668	.1854	.2058	.2238	.0992	.2042
0.8	.1294	.1440	.0626	.1308	.3258	.3552	.1350	.3402	.4682	.5004	.2052	.4836	.5190	.5526	.2780	.5352
1.0	.2612	.2816	.1124	.2750	.5940	.6324	.3018	.6262	.7602	.7884	.4408	.7816	.7960	.8200	.5304	.8042
							$\alpha = 0.01$									
0.0	.0086	.0082	.0140	.0084	.0108	.0116	.0032	.0090	.0110	.0104	.0086	.0088	.0086	.0100	.0132	.0086
0.2	.0234	.0240	.0248	.0228	.0370	.0380	.0106	.0350	.0424	.0430	.0308	.0428	.0458	.0486	.0430	.0462
0.4	.0712	.0766	.0664	.0762	.1254	.1354	.0476	.1324	.1692	.1788	.1186	.1780	.1850	.1999	.1450	.1966
0.6	.1758	.1898	.1408	.1934	.3186	.3547	.1378	.3574	.4464	.4702	.3068	.4686	.4716	.5008	.3590	.5064
0.8	.3374	.3666	.2586	.3788	.6184	.6474	.3114	.6590	.7374	.7628	.5694	.7704	.7912	.8142	.6746	.8194
1.0	.5502	.5814	.4004	.5944	.8394	.8602	.5240	.8722	.9270	.9372	.8004	.9424	.9404	.9528	.85 +	.9552
							$\alpha = 0.05$									
0.0	.0454	.0480	.0836	.0472	.0480	.0516	.0746	.0486	.0482	.0502	.0382	.0494	.0532	.0512	.0472	.0472
0.2	.0858	.0892	.1264	.0900	.1088	.1118	.1228	.1128	.1270	.1322	.0852	.1364	.1350	.1386	.1058	.1414
0.4	.1960	.2132	.2380	.2188	.3018	.3120	.2736	.3198	.3642	.3812	.2324	.3928	.3828	.4060	.2818	.4158
0.6	.3712	.4000	.3862	.4136	.5834	.6130	.5152	.6256	.6730	.7034	.4898	.7224	.7104	.7386	.5458	.7482
0.8	.5642	.6052	.5464	.6344	.8182	.8432	.7304	.8590	.8932	.9106	.7386	.9162	.9232	.9364	.8024	.9442
1.0	.7688	.7962	.7230	.8256	.9490	.9590	.8804	.9630	.9822	.9876	.9078	.9896	.9848	.9882	.9324	.9900
							$\alpha = 0.10$									
0.0	.0930	.0974	.0836	.0988	.0996	.0966	.0746	.0978	.0994	.1018	.1038	.1010	.0954	.1016	.1154	.0998
0.2	.1422	.1506	.1264	.1552	.1784	.1878	.1228	.1926	.2040	.2170	.1870	.2198	.2106	.2196	.2046	.2284
0.4	.2888	.3108	.2380	.3254	.4052	.4304	.2736	.4462	.4754	.4988	.3944	.5142	.5052	.5324	.4512	.5490
0.6	.4942	.5292	.3862	.5540	.6988	.7268	.5152	.7470	.7782	.8052	.6806	.8210	.8042	.8276	.7184	.8414
0.8	.6872	.7204	.5464	.7518	.8874	.9064	.7304	.9196	.9404	.9536	.8670	.9602	.9616	.9668	.9070	.9684
1.0	.8502	.8792	.7230	.8980	.9756	.9796	.8804	.9850	.9924	.9954	.9664	.9966	.9944	.9962	.9746	.9980

Table 2. Empirical Powers when the Underlying Distribution is CAUCHY

N_1:	70				60				50				40			
N_2:	10				20				30				40			
Shift	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher
							$\alpha = 0.001$									
0.0	.0012	.0014	.0014	.0014	.0008	.0010	.0006	.0008	.0008	.0004	.0012	.0004	.0024	.0016	.0014	.0014
0.2	.0040	.0028	.0048	.0018	.0074	.0058	.0036	.0034	.0012	.0086	.0060	.0040	.0116	.0090	.0080	.0056
0.4	.0156	.0126	.0142	.0056	.0520	.0390	.0338	.0214	.0736	.0542	.0500	.0316	.0852	.0654	.0884	.0362
0.6	.0656	.0502	.0462	.0192	.1722	.1248	.1266	.0700	.2734	.1992	.2142	.1082	.3002	.2186	.2986	.1168
0.8	.1454	.1130	.0852	.0514	.0470	.3042	.2932	.1692	.5418	.4200	.4646	.2312	.5772	.4356	.5768	.2548
1.0	.2760	.2164	.1496	.1082	.6158	.4858	.4604	.2988	.7726	.6276	.6930	.4076	.8214	.6758	.8254	.4520
							$\alpha = 0.01$									
0.0	.0082	.0084	.0146	.0078	.0096	.0086	.0034	.0084	.0096	.0084	.0112	.0068	.0112	.0110	.0134	.0090
0.2	.0264	.0252	.0372	.0186	.0458	.0398	.0230	.0280	.0534	.0432	.0626	.0342	.0596	.0508	.0754	.0366
0.4	.0896	.0754	.1014	.0482	.1870	.1502	.1130	.0982	.2384	.1888	.2604	.1242	.2634	.2150	.3198	.1422
0.6	.2194	.1860	.2246	.1148	.4238	.3392	.2744	.2258	.5468	.4516	.5794	.3154	.5772	.4660	.6540	.3280
0.8	.3922	.3210	.3464	.2156	.6832	.5744	.5290	.4244	.7986	.6896	.8150	.5258	.8354	.7186	.8838	.5362
1.0	.5506	.4716	.4586	.3424	.8432	.7422	.6950	.5908	.9348	.8544	.9362	.7052	.9509	.8878	.9684	.7354
							$\alpha = 0.05$									
0.0	.0418	.0468	.0832	.0456	.0494	.0532	.0642	.0516	.0482	.0486	.0362	.0482	.0514	.0508	.0444	.0514
0.2	.0944	.0930	.1600	.0836	.1354	.1232	.1772	.1034	.1652	.1488	.1454	.1230	.1584	.1418	.1646	.1192
0.4	.2260	.2002	.3212	.1586	.3852	.3324	.4586	.2598	.4630	.3996	.4322	.3110	.4860	.4190	.4954	.3288
0.6	.4248	.3724	.5078	.2970	.6634	.5788	.7164	.4640	.7758	.6868	.7538	.5538	.7900	.6980	.8086	.5692
0.8	.6246	.5548	.6650	.4650	.8632	.7868	.8850	.6666	.9320	.8654	.9140	.7490	.9446	.8902	.9522	.7736
1.0	.7630	.6882	.7606	.6078	.9444	.8886	.9432	.7966	.9834	.9514	.9782	.8796	.9884	.9630	.9904	.9024
							$\alpha = 0.10$									
0.0	.0928	.0966	.0832	.0970	.0934	.0960	.0642	.1006	.0974	.0988	.1082	.0996	.0948	.0972	.1182	.0996
0.2	.1680	.1636	.1600	.1422	.2132	.1998	.1772	.1796	.2504	.2314	.2760	.1980	.2438	.2278	.2972	.1976
0.4	.3352	.3104	.3212	.2618	.5030	.4482	.4586	.3696	.5782	.5224	.6136	.4300	.6090	.5394	.6692	.4472
0.6	.5400	.4910	.5078	.4200	.7636	.6918	.7164	.5932	.8550	.7890	.8754	.6782	.8726	.8092	.9072	.6898
0.8	.7303	.6710	.6650	.5838	.9222	.8602	.8850	.7724	.9650	.9218	.9710	.8384	.9730	.9362	.9802	.8598
1.0	.8472	.7818	.7606	.7218	.9714	.9334	.9432	.8738	.9936	.9754	.9932	.9320	.9974	.9826	.9978	.9406

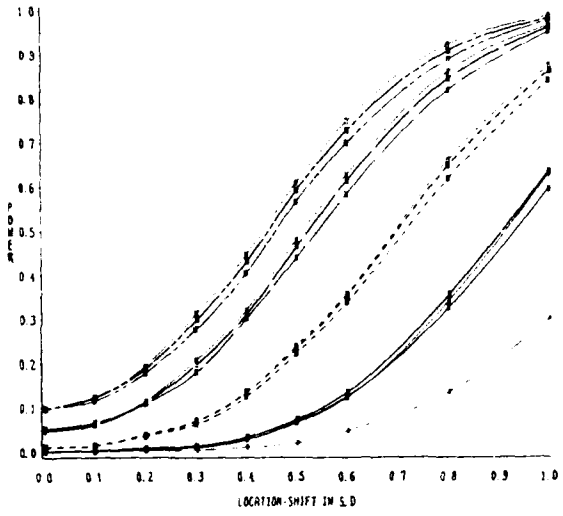
Table 3. Empirical Powers when the Underlying Distribution is LAPLACE

N_1:	70				60				50				40			
N_2:	10				20				30				40			
Shift	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher	δ_1	δ_2	Median	Fisher
							$\alpha = 0.001$									
0.0	.0016	.0016	.0012	.0008	.0006	.0008	.0004	.0002	.0014	.0010	.0002	.0008	.0006	.0008	.0002	.0006
0.2	.0046	.0032	.0062	.0020	.0108	.0086	.0048	.0048	.0176	.0154	.0092	.0106	.0146	.0122	.0142	.0078
0.4	.0300	.0270	.0194	.0170	.0834	.0696	.0512	.0468	.1202	.1088	.0720	.0722	.1368	.1160	.1058	.0790
0.6	.1080	.0146	.0602	.0590	.2938	.2624	.1688	.1830	.4158	.3740	.2879	.2768	.4534	.4062	.3860	.3070
0.8	.2632	.2440	.1300	.1748	.5880	.5478	.3880	.4452	.7546	.7168	.5912	.5932	.7934	.7502	.6956	.6368
1.0	.4772	.4480	.2242	.3580	.8358	.8040	.6018	.7152	.9322	.9128	.8172	.8438	.9578	.9408	.9088	.8732
							$\alpha = 0.01$									
0.0	.0096	.0104	.0124	.0098	.0102	.0084	.0046	.0058	.0106	.0104	.0102	.0092	.0116	.0102	.0136	.0100
0.2	.0324	.0290	.0416	.0248	.0558	.0498	.0254	.0404	.0748	.0684	.0810	.0554	.0732	.0688	.0894	.0550
0.4	.1174	.1118	.1232	.0882	.2416	.2204	.1442	.1802	.3204	.2910	.3174	.2434	.3592	.3280	.3896	.2672
0.6	.3164	.2944	.2772	.2454	.5836	.5452	.3662	.4664	.7042	.6684	.6736	.5852	.7342	.6978	.7340	.6048
0.8	.5442	.5244	.4402	.4706	.8382	.8090	.6276	.7410	.9246	.8998	.8942	.8474	.9432	.9262	.9356	.8742
1.0	.7546	.7290	.6008	.6824	.9628	.9480	.8126	.9180	.9884	.9824	.9724	.9644	.9934	.9890	.9904	.9750
							$\alpha = 0.05$									
0.0	.0450	.0484	.0880	.0478	.0486	.0490	.0646	.0490	.0450	.0450	.0350	.0450	.0522	.0528	.0464	.0518
0.2	.1100	.1124	.1698	.1014	.1612	.1544	.2054	.1394	.1970	.1882	.1706	.1720	.1986	.1930	.1904	.1726
0.4	.2770	.2746	.3610	.2504	.4636	.4420	.5090	.3982	.5516	.5268	.5016	.4718	.5964	.5702	.5772	.5130
0.6	.5526	.5362	.5854	.4916	.7856	.7590	.7896	.7166	.8722	.8552	.8252	.8070	.8918	.8708	.8624	.8252
0.8	.7530	.7376	.7436	.7096	.9450	.9304	.9312	.9040	.9802	.9720	.9580	.9538	.9856	.9806	.9736	.9646
1.0	.9002	.8880	.8638	.8768	.9930	.9884	.9852	.9780	.9984	.9976	.9920	.9942	.9996	.9988	.9974	.9962
							$\alpha = 0.10$									
0.0	.0952	.1006	.0880	.1028	.0976	.0990	.0696	.0948	.0914	.0962	.1024	.0980	.1006	.1040	.1174	.1058
0.2	.1812	.1804	.1698	.1790	.2486	.2420	.2054	.2264	.2924	.2874	.3214	.2652	.3022	.2846	.3404	.2654
0.4	.3910	.3886	.3610	.3638	.5870	.5698	.5090	.5264	.6750	.6538	.6838	.6032	.7104	.6902	.7396	.6484
0.6	.6556	.6550	.5854	.6234	.8650	.8496	.7896	.8142	.9294	.9186	.9178	.8858	.9374	.9280	.9420	.8954
0.8	.8340	.8250	.7436	.8140	.9718	.9636	.9312	.9496	.9904	.9872	.9872	.9872	.9942	.9910	.9916	.9826
1.0	.9458	.9352	.8638	.9268	.9976	.9956	.9852	.9914	.9994	.9996	.9978	.9976	1.000	.9998	.9996	.9992

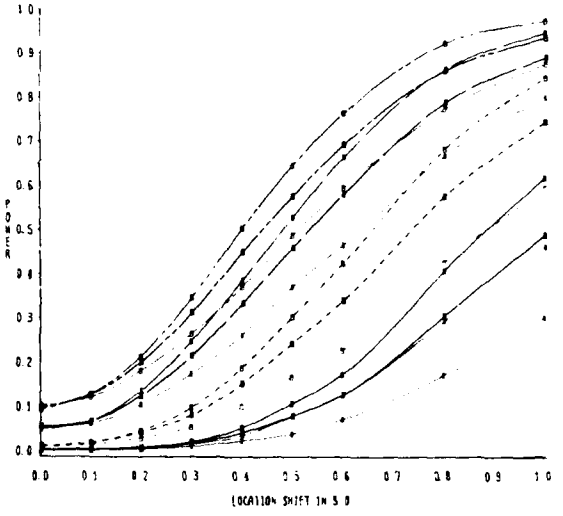
POWER OF MRPP TESTS

UNDERLYING DISTRIBUTION:

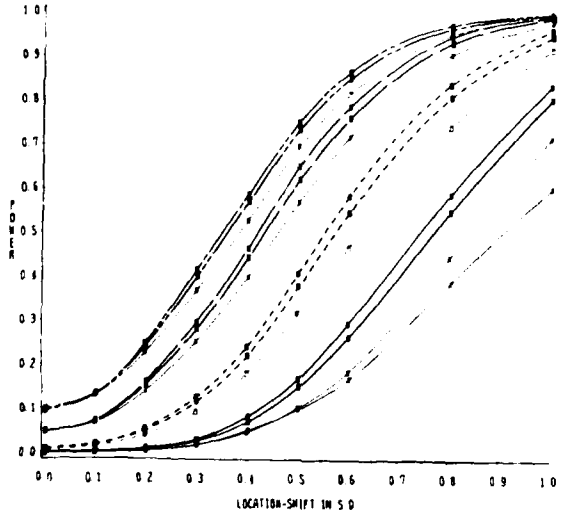
NORMAL N1=60, N2=20



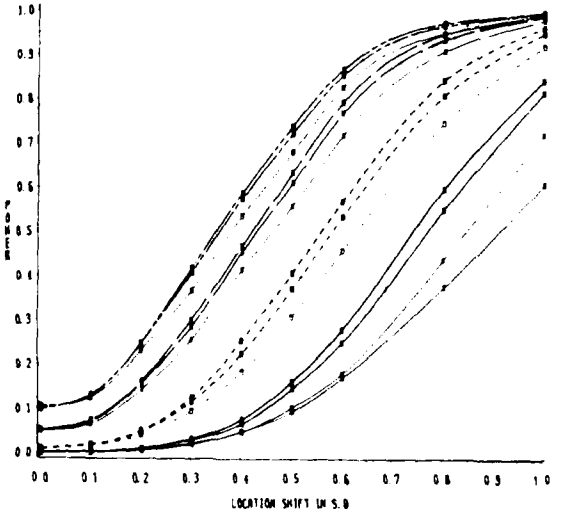
CAUCHY N1=60, N2=20



LAPLACE N1=60, N2=20



LAPLACE N1=20, N2=60

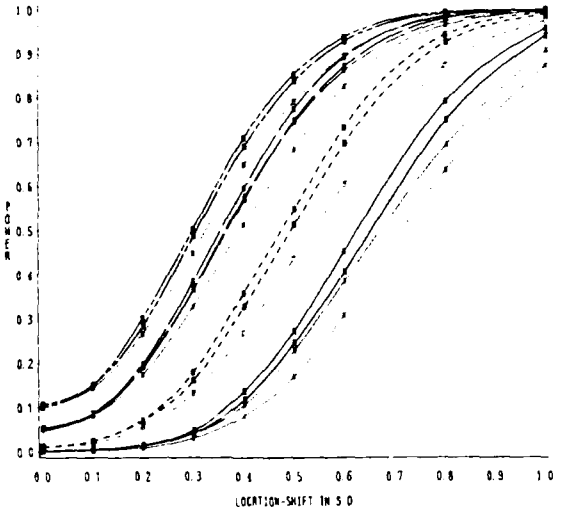


α=0.001=———, α=0.01=– – – –
α=0.05 =— — —, α=0.10=— – — –

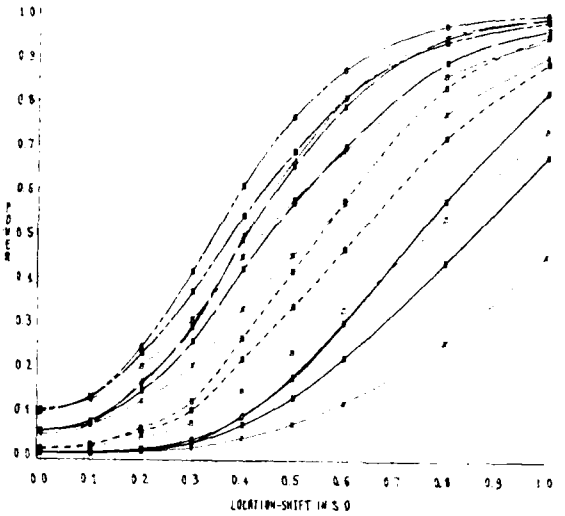
POWER OF MRPP TESTS

UNDERLYING DISTRIBUTION·

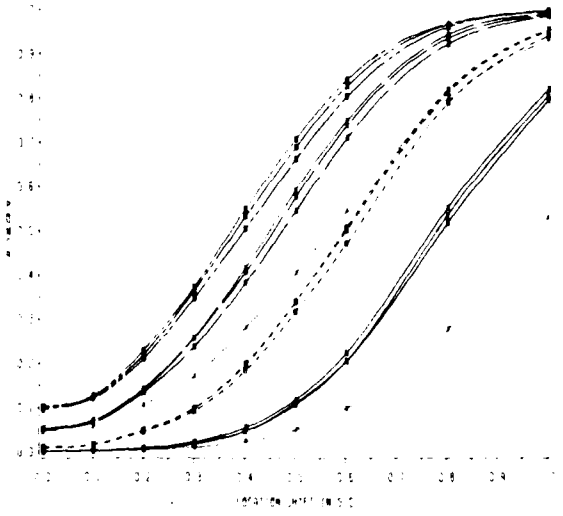
LAPLACE N1=40, N2=40



CAUCHY N1=40, N2=40



NORMAL N1=40, N2=40



α=0.001 = ——, α=0.01 = – – –
α=0.05 = ——, α=0.10 = —— – – ——

References

Brockwell, P.J., Mielke, P.W. and Robinson, J. (1982). On non-normal invariance principles for multiresponse permutation procedures. Austral. J. Statist. 24, 33-41.

Mielke, P.W. (1984). Meteorological applications of permutation techniques based on distance functions. In P.R. Krishnaiah and P.K. Sen, eds., Handbook of Statistics, vol. 4, North-Holland, Amsterdam, 813-830.

Mielke, P.W., Berry, K.J., Brockwell, P.J. and Williams, J.S. (1981). A class of nonparametric tests based on multiresponse permutation procedures. Biometrika 68, 720-724.

Mielke, P.W., Berry, K.J. and Medina, J.G. (1982). Climax I and II: Distortion resistant residual analyses. J. Appl. Meteor. 21, 788-792.

Tracy, D.S. and Tajuddin, I.H. (1985). Extended moment results for improving inferences based on MRPP. Comm. Statist. - Theor. Meth. 14, 1485-1496.

Tracy, D.S. and Tajuddin, I.H. (1986). Empirical power comparisons of two MRPP rank tests. Comm. Statist. - Theor. Meth. 15, 551-570.

Performance of Several One Sample Procedures

David L. Turner and YuYu Wang, Utah State University

Abstract

Empirical p-values and powers for the usual t test, the signed rank test, a trimmed t test, a jackKnife and a bootstrap procedure were compared using repeated samples of size 30 from normal, double exponential, Cauchy, negative exponential and uniform distributions for normal power values ranging from 0.05 through 0.95. The Bootstrap performed as well as the usual t test. The trimmed t, signed rank test and the usual t-test performed about the same. The jackknife performed worst among these tests. The signed rank test did best for the Cauchy distribution.

1. Introduction.

The JackKnife and Bootstrap procedures as described by Efron(1982) seem to place an inordinate amount of emphasis on the sample and how well it approximates the true underlying cumulative distribution function (cdf). To investigate the performance of these two techniques relative to some more standard or usual tests, a monte carlo study was run. Table 1 lists the 5 different test statistics and methods used to compute p-values for each test. Samples were generated from a standard normal, a double exponential, negative exponential, uniform and Cauchy distributions. Each started with uniform deviates generated by the portable congruential random number generator given by Wichmann and Hill(1987). Each distribution was scaled to have mean 0 and variance 1. The Cauchy was scaled to have zero median and the same interquartile range as the standard normal.

100 trials were run for each combination of μ_{true}, distribution and sample size. After some preliminary runs, 100 bootstrap samples were deemed adequate for the demonstration purposes of this paper.

2. p-value Analysis for H_o: $\mu = \mu_o$ true

The first analyis focused on the case when the null hypothesis was true, i.e. when μ was indeed equal to μ_o. Initially the t distribution was used to calculate p-values for the jackKnife and bootstrap procedures. This consistently gave average p-values slightly larger than 0.5. A more "nonparametric" p-value was then implemented for these 2 procedures which defined p as $2[\min(\#\bar{Y}'s < \mu_o, \#\bar{Y}'s > \mu_o)]$.

Since the null hypothesis was in fact true, the p-values should have followed a uniform distribution. Figure 1 plots the empirical cdf of the p-values against the cdf for a uniform distribution for samples from each of the distributions when n = 30. A 45° line for these plots was regraded as the standard, indicating no departure from the underlying model.

p-values for the Cauchy distribution tend to clump or cluster in the middle a little more than they should for the t and the trimmed t. The negative exponential also has "light tails" for the signed rank test. The bootstrap comes surprisingly close to the optimal 45° line, indicating that the p-values for a true null hypothesis are close to uniformly distributed. The p-values for the jackKnife show a completly different story however. It appears to be very difficult to get a jackknifed $\bar{Y}_j$ more extreme than the original as indicated by the fact that far more small p-values were observed than expected.

3. p-value Analysis for H_o: $\mu = \mu_o$ false.

The next step in the analysis involved specifying values for the μ_{true} to make the power of the t-test take on the values 0.05, .10, .25, .50, .75, .90 and .95 for n = 10 . Figures 2 and 3 plot the average p-values for 100 repetitions of each μ_{true} value for each distribution. Figure 2 shows how poorly the jackknife does across the 5 distributions considered here. It is surprising to see how little fluctuation there is among the tests for samples from all but the Cauchy distribution. Each of the tests except the Jackknife seems to be fairly robust to departures from normality. The signed rank test is the clear winner for the Cauchy distribution, having a curve more like those for sampling from the normal distribution. Figure 3 plots the same values but with a different arrangement. Each test seems to do poorly for the Cauchy distribution except the signed rank test.

4. Empirical power analysis

The null hypothesis for this study was rejected for any particular run if the empirical p-value was less than α = 0.05. Figures 4 and 5 plot the empirical power values for the 5 tests against the 5 distributions plotted against the 7 different μ values. Figure 4 shows very close agreement

among all tests but the jackknife for all but the Cauchy distribution. For the Cauchy, the signed rank test comes closest to providing an "ideal" power curve. The bootstrap shows a slight improvement over the t test, but does not do as well as the trimmed t. The jackknife almost always rejects, and seems to differ but little as the value of μ_{true} changes. Figure 5 shows that all but the signed rank test provide very low power for the Cauchy distribution.

5. Summary and Conclusion

For the 5 distributions considered here, the bootstrap did surprisingly well as did the signed rank test. Equally surprising or comforting was how well the t test did. For these 5 distributions, the usual t appears to be robust enough. The present formulation of the jackknife is not useful, being far too liberal and rejecting far more often than it should.

Further research in this area could include extreme value distributions and perhaps a mixture distribution such as $0.9N(0,1) + 0.1N(0,100)$, or more "L" shaped distributions. More complex procedures could also be tried.

References

Efron, Bradley (1982): The Jackknife, the Bootstrap and Other Resampling Plans, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Wichmann, B.A. and I.D. Hill (1987): "Building a Random-Number Generator," Byte Volume 12, Number 3, pp 127-128.

Table 1. One sample procedures compared.

Procedure	Test Statistic	p-value calculation
t statistic	$(\bar{Y} - \mu_o)/(S/\sqrt{n})$	t distribution
Trimmed t	t for trimmed data (deleting lower 5% and upper 5%)	t distribution
Signed Rank t	t for signed ranks of $(Y_i - \mu_o)$	t distribution
Jackknife	$\bar{Y}_i$'s computed by deleting each Y_i	$2[\min(\#\bar{Y}'s < \mu_o, \#\bar{Y}'s > \mu_o)]$
Bootstrap	$\bar{Y}_i$'s computed from random samples with replacement from the data	$2[\min(\#\bar{Y}'s < \mu_o, \#\bar{Y}'s > \mu_o)]$

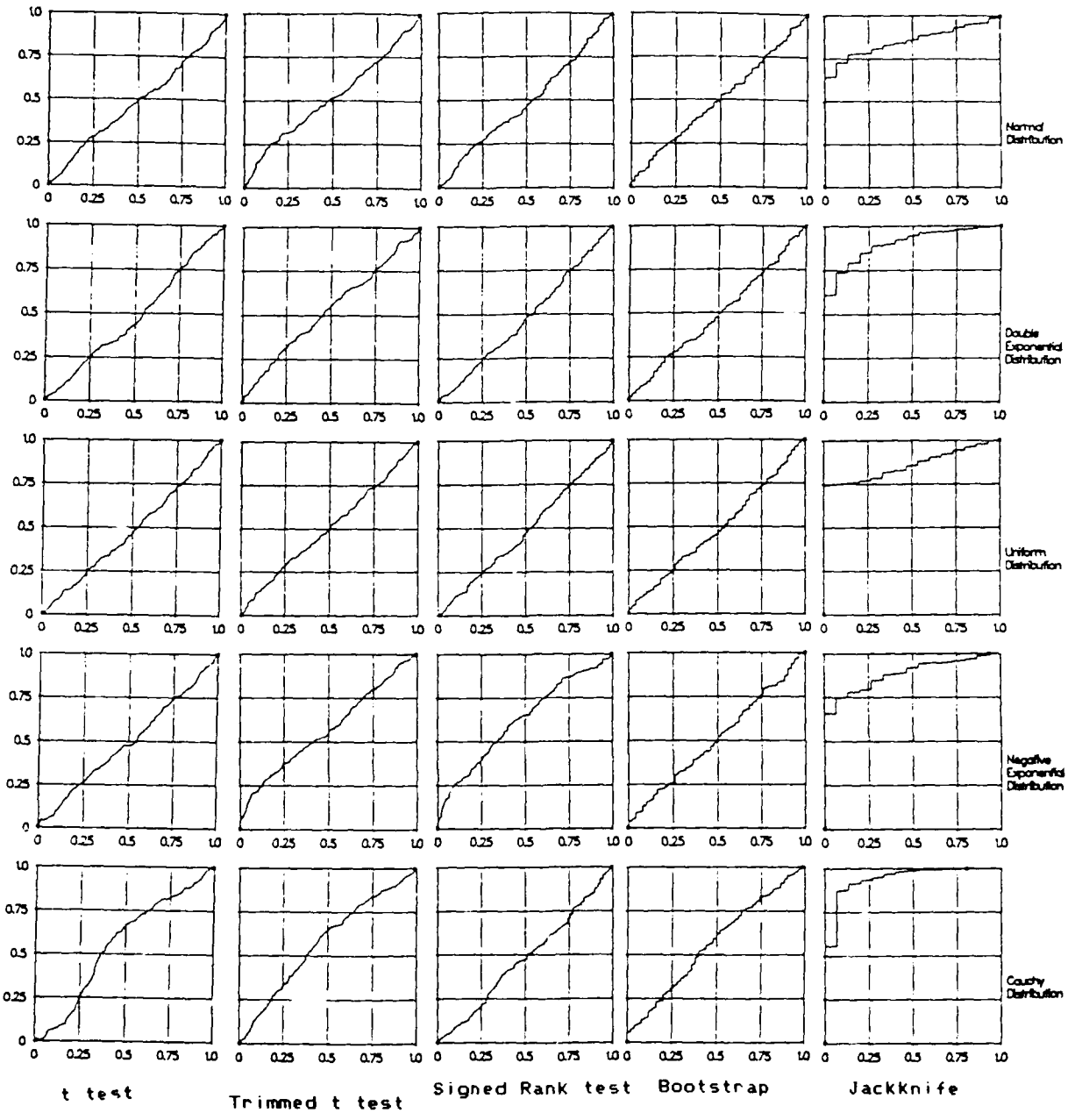


Figure 1. Plots of the empirical cdf of the p-values against the uniform cdf for each test and distribution for n = 30.

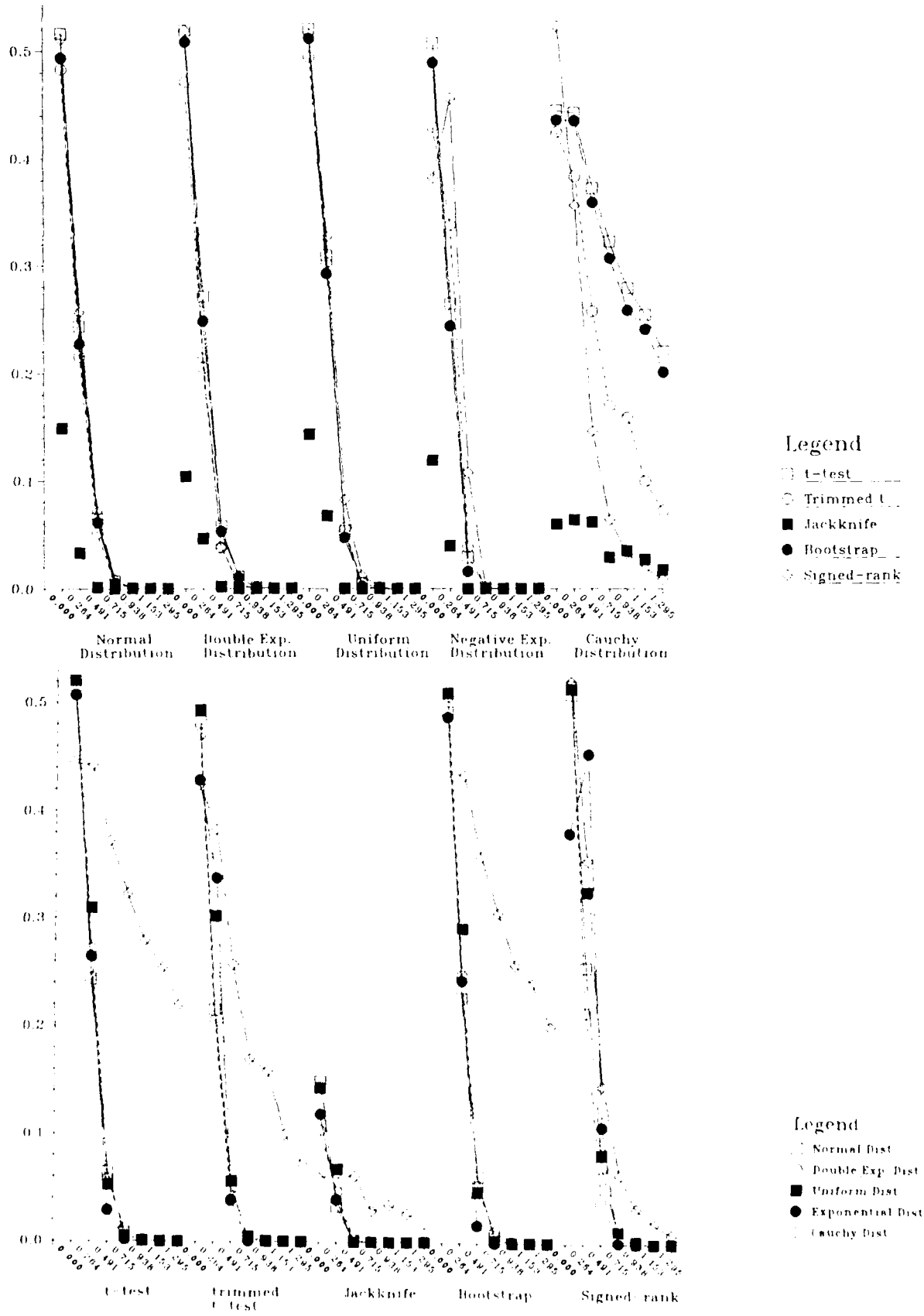


Figure 2. Plots of the average p-values for n = 30 for each distribution and test combination for false H_o's.

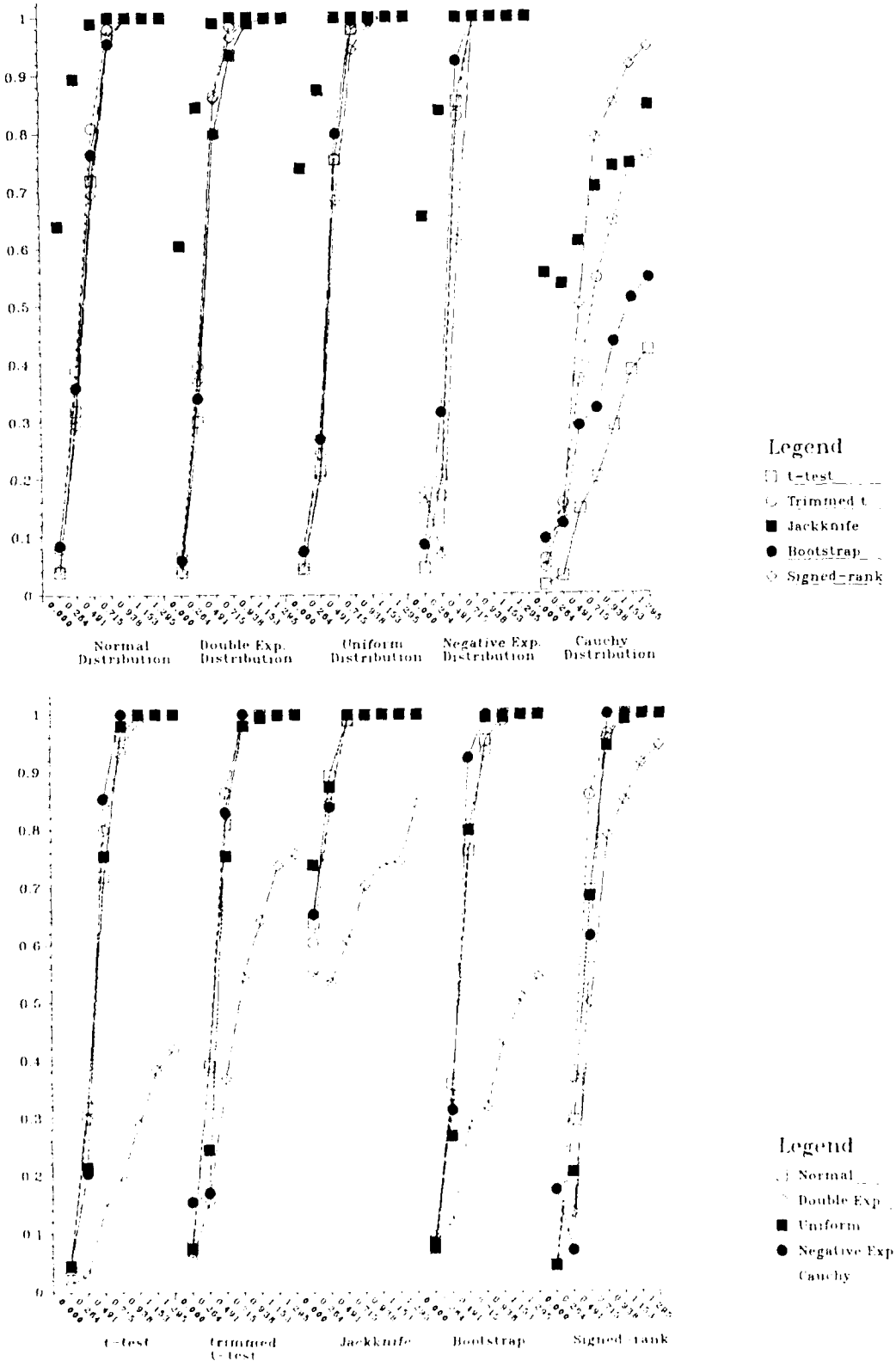


Figure 3. Plots of the average p-values for n = 30 for each test and distribution combination for false H_o's.

XV. TIME SERIES ANALYSIS

COMPUTATIONAL ASPECT OF HARMONIC SIGNAL DETECTION

Keh-Shin Lii, University of California, Riverside and Tai-Houn Tsou, University of California, Riverside

1. INTRODUCTION

We consider a model of the form

$$X_t = Y_t + Z_t \tag{1}$$

with Y_t a periodic function given by

$$Y_t = \sum_{k=1}^{K} A_k \exp(-it\omega_k) = \sum_{k=1}^{K} R_k \cos(\omega_k t + \phi_k) \tag{2}$$

where R_k, ω_k, and ϕ_k are the amplitude, frequency and phase of the harmonic process Y_t, Z_t is an additive noise process which is independent of Y_t. When Z_t is white, Schuster (1898), Fisher (1929), Whittle (1952), and Siegel (1980) discussed how to detect the harmonic signal Y_t. In many applications of engneering, meteorology and ecology problems, the background noise may not be white, or it can be represented as a linear process, such as

$$Z_t = \sum_u \alpha_u \varepsilon_{t-u} \tag{3}$$

where ε_t's are independent, identically distributed and α_u's are constants. Usually we assume that ε_t has mean zero with variance σ_ε^2. Whittle (1954), Bartlett (1957), and Priestley (1962 a,b) dealt with the testing and estimation problems, when the noise process is assumed to be colored.

To motivate our procedure, we first consider the Fisher's test. This is the case when the noise process Z_t is assumed to be zero mean Gaussian white noise with variance σ^2. The null hypothesis can be stated as

H_0 : the harmonic signal Y_t is zero in (1).

Under H_0, the periodogram of the process X_t, $I_N^X(\lambda_j) = (2/N)\left|\sum_{t=1}^{N} X_t \exp(-it\lambda_j)\right|^2$ with $\lambda_j = 2\pi j/N$ has a Chi-square distribution with 2 degrees of freedom, if it is divided by σ^2. Furthermore, $I_N^X(\lambda_j)$ and $I_N^X(\lambda_k)$ are independent if $j \neq k$ with $\lambda_m = 2\pi m/N$ for $j,k = 1,2,\ldots,[N/2]$. This result also holds asymptotically when the noise process Z_t is independent, identically distributed, but not necessarily Gaussian [Brillinger (1975) p.94]. Based upon the previous result, Fisher (1929) derived the exact distribution for the test of the largest peak of the periodogram, i.e.

$$G^{(f)} = \frac{\max_{1 \le j \le [N/2]} I_N^X(\lambda_j)}{\sum_{1 \le j \le [N/2]} I_N^X(\lambda_j)} . \tag{4}$$

When the noise process is linear which has the form given in (3) with the conditions that $E|\varepsilon_t|^4 < \infty$, and $\sum |\alpha_u||u| < \infty$, then the power spectrum of Z_t is

$$f^Z(\lambda) = (\sigma_\varepsilon^2/2\pi)\left|\sum_u \alpha_u \exp(-iu\lambda)\right|^2$$

and its periodogram has the following relationship with the periodogram of ε_t

$$I_N^Z(\lambda) = \left|\sum \alpha_u \exp(-iu\lambda)\right|^2 I_N^\varepsilon(\lambda) + R_N(\lambda)$$

where $R_N(\lambda) = O(1/N)$ uniformly in λ, [Priestley (1981) p.424]. From this, it is clear that asympototically

$$\frac{I_N^Z(\lambda)}{\left|\sum \alpha_u \exp(-iu\lambda)\right|^2} \approx I_N^\varepsilon(\lambda) \tag{5}$$

if $\left|\sum \alpha_u \exp(-iu\lambda)\right| \neq 0$ for all λ. The asymptotic distribution of $I_N^\varepsilon(\lambda)$ is known from the previous discussion. This observation motivated various methods which attempt to estimate the spectrum of the noise process Z_t which is proportional to $\left|\sum \alpha_u \exp(-iu\lambda)\right|^2$ under the null hypothesis H_0. We will first review some conventional methods which include Whittle's test, Bartlett's test and Priestley's P(λ) test.

(I) Whittle's test - Whittle (1952, 1954)

The basic idea of this approach is to use the asymptotic relationship between the periodogram of the general linear process $\{Z_t\}$ and that of the residual process $\{\varepsilon_t\}$. Following Fisher's tests in equation (4), Whittle proposed the test statistic

$$G^{(w)} = \frac{\max_j \{I_N^X(\lambda_j)/2\pi f^Z(\lambda_j)\}}{\sum_{j=1}^{n} \{I_N^X(\lambda_j)/2\pi f^Z(\lambda_j)\}} . \tag{6}$$

where $j = 1,\ldots,n$, $n = [N/2]$.

Under H_0, $G^{(w)}$ is asymptotically distributed as $G^{(f)}$. The problem is, the actual spectral density function $f^Z(\lambda)$ of $\{Z_t\}$ is usually unknown. The remedy for this is to use the estimated power spectrum of $\{Z_t\}$ as a substitute for $f^Z(\lambda_j)$.

(II) Grouped periodogram test - Bartlett (1957)

This method divides the periodogram ordinates into $r = [N/2k]$ sets, each set containing k ordinates. When k is relatively small compared with N, the spectral density function in this region is almost flat. Thus, on the frequency domain, if we choose the bandwidth of the smoothing kernel small enough, the estimated power spectrum of $\{Z_t\}$ will be almost constant for these k ordinates, except the harmonic terms in the frequency ω. Let

$$G_k^{(g)} = \max_{(r-1)k+1 < j < rk} I_N^X(\lambda_j) \Big/ \sum_{j=(r-1)k+1} I_N^X(\lambda_j), \tag{7}$$

under H_0, $G_k^{(g)}$ has approximately the same distribution as Fisher's test with k degrees of freedom.

(III) P(λ) test - Priestley (1962 a,b)

The idea behind this test is to use properties

of the autocovariance function (ACF) of Y_t and Z_t. It is assumed that the ACF of the general linear process Z_t will die out as lag $u \to \infty$, and the ACF of the harmonic process Y_t will persist even for large u. Let

$$P(\lambda) = (1/2\pi)\sum_{u=-N+1}^{N-1}(K_n^{(1)}(u) - K_m^{(2)}(u))\hat{C}(u)\exp(-i\lambda u) = f_n^X(\lambda) - f_m^X(\lambda),$$

where $\hat{C}(u)$ is the sample autocovariance function and $K_n^{(1)}(u)$, $K_m^{(2)}(u)$ are two symmetric sequences of weight function such that both decrease as $|u|$ increases and $m/n \to 0$, $n/N \to 0$, as $m \to \infty$, $n \to \infty$ and $N \to \infty$.

To detect the harmonic process, we plot $P(\lambda)$ vs. λ and test the significance of the large peaks. The standardized cumulative sums can be defined as

$$J_q = \frac{\sqrt{N}/(m\Lambda_{n,m})\sum_{j=1}^{q} P(2\pi j/m)}{\{(1/2\pi)G(\pi)\}^{1/2}}, \qquad (8)$$

where $q = 0,1,\ldots,[m/2]$, and $G(\pi)$ is estimated by $G^*(\pi) = (1/4\pi)\Sigma_{u=-m}^{m}\hat{C}^2(u)$. Detail of this discussion can be found in Priestley's (1981).

All these methods are based on the second order moments of the time series. When the noise process is Gaussian then moments up to second order give all the information. If the noise process is non-Gaussian then cumulants of order greater than two might provide extra information in addition to those of less than or equal to second order moments. In next section we will present a method which will take advantage of third and fourth order cumulants to improve the efficiency of detecting the existence of the periodic function Y_t under the assumption that the noise process is non-Gaussian.

2. TEST STATISTICS

For simplicity, we will study processes, Y_t and Z_t separately. Assuming X_t is stationary up to order eight, and all cumulants are summable up to the eight order. The bispectrum and the trispectrum is the Fourier transformation of third and fourth order cumulant function, i.e.

$$f^X(\lambda_1,\lambda_2) = (2\pi)^{-2}\sum_{u,v=-\infty}^{\infty} C^X(u,v)\exp(-iu\lambda_1 - iv\lambda_2) = f^Y(\lambda_1,\lambda_2) + f^Z(\lambda_1,\lambda_2)$$

$$f^X(\lambda_1,\lambda_2,\lambda_3) = (2\pi)^{-3}\sum_{u,v,s=-\infty} C^X(u,v,s)\exp(-iu\lambda_1 - iv\lambda_2 - is\lambda_3) = f^Y(\lambda_1,\lambda_2,\lambda_3) + f^Z(\lambda_1,\lambda_2,\lambda_3). \qquad (9)$$

Assume that the harmonic process $\{Y_t\}$ contains only one harmonic component, i.e.

$$Y_t = R\cos(\omega t + \phi) \qquad \phi \sim U(-\pi,\pi).$$

Let $\delta_N(\theta) = \sin((N+(1/2))\theta)/2\pi\sin(\theta/2)$ be the Dirichlet kernel. Then, on the particular submanifold $(\lambda,0)$ and $(\lambda,0,0)$, we have

$$I_N^Y(\lambda,0) = O(1/N)$$

$$I_N^Y(\lambda,0,0) = (3R^4/16)[\delta_N(\omega-\lambda) + \delta_N(\omega+\lambda)]\delta_N^2(\omega). \qquad (10)$$

Now consider the noise process $\{Z_t\}$. When $\{Z_t\}$ is non-Gaussian linear process with 3rd (4th) order cumulants γ_3 (γ_4) of ε_t exist, then the bispectrum and the trispectrum can be represented as

$$f^Z(\lambda_1,\lambda_2) = (\gamma_3/(2\pi)^2)\Gamma(\lambda_1)\Gamma(\lambda_2)\Gamma^*(\lambda_1+\lambda_2)$$

$$f^Z(\lambda_1,\lambda_2,\lambda_3) = (\gamma_4/(2\pi)^3)\Gamma(\lambda_1)\Gamma(\lambda_2)\Gamma(\lambda_3)\Gamma^*(\lambda_1+\lambda_2+\lambda_3)$$

with $\Gamma(\lambda) = \Sigma\,\alpha_u\exp(-iu\lambda)$.

When the noise process Z_t is given in (3) such that the third order cumulant of ε_t is nonzero, the bispectrum of the process X_t, on the submanifold $(\lambda_1,\lambda_2) = (\lambda,0)$, can be represented as

$$f^X(\lambda,0) = f^Z(\lambda,0) = (2\pi)^{-2}\gamma_3\Gamma(0)|\Gamma(\lambda)|^2.$$

Hence, when $\Gamma(0) \neq 0$,

$$|\Gamma(\lambda)|^2 = D_1 f^X(\lambda,0)$$

with $D_1 = (2\pi)^2/(\gamma_3\Gamma(0))$. From these discussions the following test statistics, from (5), is proposed

$$G^{(b)} = \frac{\max_{1<j<[N/2]} I_N^X(\lambda_j)/Rf_N^X(\lambda_j,0)}{\sum_{1<j<[N/2]} I_N^X(\lambda_j)/Rf_N^X(\lambda_j,0)} \qquad j=1,2\ldots,[N/2] \qquad (11)$$

where $Rf_N^X(\lambda,0)$ is a consistent estimator of the real part of the bispectrum $f^X(\lambda_1,\lambda_2)$ at frequency $(\lambda,0)$, and the unknown constant D_1 is cancelled out.

When ε_t is symmetric distributed or $\gamma_3 = 0$ with nonzero 4th order cumulant γ_4, equation (9) implies

$$D_2 f^X(\lambda,0,0) = D_2 f^Y(\lambda,0,0) + |\Gamma(\lambda)|^2$$

where $D_2 = (2\pi)^3/\gamma_4\Gamma^2(0)$. According to (10), the bias around the harmonic frequency using $D_2 f_N^Y(\lambda,0,0)$ is generally smaller than using $2\pi/\sigma_\varepsilon^2 f_N^Y(\lambda)$ due to smoothing. Thus, the trispectrum of X_t provide an estimate of $|\Gamma(\lambda)|^2$ which has smaller bias than the methods using the power spectrum. The following statistics are used to detect the harmonic component

$$G^{(t)} = \frac{\max_{1<j<[N/2]} I_N^X(\omega_j)/Rf_N^X(\lambda_j,0,0)}{\sum_{\substack{1\le j\le[N/2]\\ j=1,2\ldots,[n/2]}} I_N^X(\omega_j)/Rf_N^X(\lambda_j,0,0)} \qquad (12)$$

where $Rf_N^X(\lambda_j,0,0)$ is the estimated real part of the trispectrum at $(\lambda_j,0,0)$.

3. ESTIMATION AND COMPUTATION OF NEW STATISTICS

Since the bispectrum and trispectrum are

defined as (9), their natural estimates are

$$I_N^X(\lambda_1,\lambda_2) = 1/(2\pi)^2 \sum_{u,v=-N+1}^{N-1} \hat{C}^X(u,v)\exp(-i\lambda_1 u - i\lambda_2 v),$$

$$I_N^X(\lambda_1,\lambda_2,\lambda_3) = (2\pi)^3 \sum_{u,v,s=-N+1}^{N-1} \hat{C}^X(u,v,s)\exp(-i\lambda_1 u,-i\lambda_2 v - i\lambda_3 s)$$

respectively, where

$$\hat{C}^X(u,v) = m^X(u,v),$$

$$m^X(u,v) = (1/T_1) \sum_{t=-\min(u,v,0)+1}^{N-\max(u,v,0)} X_t X_{t+u} X_{t+v}$$

with $-N+1 \le u,v \le N-1$,

$T_1 = N - \max(u,v,0) + \min(u,v,0)$ and

$$\hat{C}^X(u,v,s) = m^X(u,v,s) - m^X(u)m^X(v-s) - m^X(v)m^X(u-s) - m^X(s)m^X(u-v),$$

$$m^X(k-h) = (1/T_2) \sum_{t=-\min(u,v,s,0)+1}^{N-\max(u,v,s,0)} X_{t+h} X_{t+k}$$

$$m^X(u,v,s) = (1/T_2) \sum_{t=-\min(u,v,s,0)+1}^{N-\max(u,v,s,0)} X_t X_{t+u} X_{t+v} X_{t+s}$$

$-N+1 \le u,v,s, \le N-1$,

where $T_2 = N - \max(u,v,s,0) + \min(u,v,s,0)$ and (h,k) is any two elements partition of the set $(0,u,v,s)$.

Noticed that the third and fourth order periodogram are not consistent estimator of bispectrum and trispectrum. Rosenblatt and Van Ness (1965), Billinger and Rosenblatt (1967), mentioned two different approaches to estimate the bispectrum and trispectrum consistently. One way by Fourier transform the smoothed 3rd and 4th cumulant function, the other by smoothing the 3rd and 4th order periodogram function.

The advantages for the later approach are reducing the computational time and saving the computer storage. Once the Fourier transformation of the random process is given, the rest of calculations for 3rd and 4th order periodogram is just the multiplication on different frequencies. The disadvantage is that it fails to provide the bispectrum and trispectrum on the submanifolds. For the submanifolds, Billinger and Rosenblatt (1967) suggested averaging values in a neighborhood of a submanifold to approximate the exact calculation. In contrast, the former approach requires larger computer memory and longer computational time to calculate the 3rd and 4th order periodogram but it provides direct estimates of bispectrum and trispectrum for all frequencies including those of the submanifold.

Since we are only interested in the bispectrum and trispectrum on certain submanifolds, we then focus on the time domain smoothing method only. The estimate of bispectrum and trispectrum can be obtained by

$$f_N^X(\lambda_1,\lambda_2) = (2\pi)^{-2} \sum_{u,v=-N+1}^{N-1} K_{M_1}(u,v)\hat{C}^X(u,v) \exp(-iu\lambda_1 - iv\lambda_2) \quad (13)$$

$$f_N^X(\lambda_1,\lambda_2,\lambda_3) = (2\pi)^{-3} \sum_{u,v,s=-N+1}^{N-1} K_{M_2}(u,v,s) \hat{C}^X(u,v,s)\exp(-iu\lambda_1 - iv\lambda_2 - is\lambda_3) \quad (14)$$

where $K_{M_1}(u,v) \sim K(u/M_1, v/M_1)$ and $K_{M_2}(u,v,s) \sim K(u/M_2, v/M_2, s/M_2)$ are 2- and 3- dimensional lag window with sequence of constants $\{M_1\}$ and $\{M_2\}$, which tends to infinity ad $N \to \infty$, and $M_1^2/N \to 0$ $M_2^3/N \to 0$. A easy way to create the 2- and 3-dimensional lag window is taking the product of one dimensional lag window. Based on equation (13) and some assumptions, Rosenblatt and Van Ness (1965), actually derived the mean and variance of bispectrum estimate which are given as theorem 4 and theorem 5 in their paper. More general results including trispectrum are given by Lii and Rosenblatt (1988). From these discussions and equations (13) and (14), we have consistent estimates of power spectrum, up to a constant, from the estimate of the bispectrum $f_N^X(\lambda,0)$ and the estimate of the trispectrum $f_N^Y(\lambda,0,0)$. These consistent estimates are used in the test statistics $G^{(b)}$ and $G^{(t)}$ given in equations (11) and (12).

4. SIMULATION RESULT

To demonstrate the effectiveness of the $G^{(b)}$ and $G^{(t)}$ test in equation (11) and (12), we will now study two simulation series which have mixed spectra. Consider the simulated series from equation (1) with k=1 for harmonic process Y_t and and linear process Z_t defined as follow

$$Y_t = R\cos(\omega t)$$

$$Z_t + 1.2Z_{t-1} + 0.6Z_{t-2} = \epsilon_t$$

where the coefficient of AR(2) process Z_t are $\phi_1=1.2$, $\phi_2=0.6$, and ϵ_t are independent exponentially distributed random deviates with mean one generated from the GGEXN subroutine in the IMSL. The number of observations generated for each series is N=256. Different values for R and ω are used to compare the power of the methods under different conditions. We choose R = 0.5, $\omega/2\pi = 1/4$ in series 1 and R = 1.0, $\omega/2\pi = 27/64$ in series 2. The process of X_t, and its periodogram are shown in Figure 1 and Figure 2.

Results for different testing methods are presented in the followings

(i) Whittle's test

Here we select the Bartlett window to be the smoothing function of Z_t with truncation parameter M=25 based on the autocovariance function. Since

$$P(G^{(W)} > z) \approx 1-(1-\exp(-nz))^n, \quad n=[N/2]-1,$$

thus, if we choose the significant level α = .1, .05, and .01 from $z_\alpha = -\ln(1-(1-\alpha)^{1/n})/n$, we have $z_{.1}$ = .0559, $z_{.05}$ = .0615, and $z_{.01}$ = .0743 respectively. Figure 3 presents the $I_N^x(\lambda_j)/f_N^x(\lambda_j)$ plot vs. frequency which shows a number of suspected large peaks. These peaks are used to test the existence of harmonic components.

The final result is presented in Table 1, where *, ** and *** indicate that the Whittle's $G^{(W)}$ statistics are significant at α level .1, .05, and .01 respectively. From Table 1, we find that Whittle's test has difficulties in detecting the harmonic components when the mass spectrum of the harmonic signal $f^y(\omega_k)$ is mixed with the large spectrum of the noise $f^z(\lambda_j)$, such as the series 2.

(ii) Bartlett's test

Since the grouping parameter k is selected arbitariliy, for comparison purposes, we use the test with (a) k=4 (b) k=8 (c) k=12. The critical value z for $G_k^{(g)}$ can be calculated by approximating Fisher's $G^{(f)}$ with k degrees of freedom, thus, we have $z_\alpha = 1-(\alpha/n)^{1/(k-1)}$. Table 2 shows z_α value for different grouping parameter. Table 3 summarizes the results in which one can see that Bartlett's test is similar as Whittle's test, it detects the periodicity in series 1 but fails to detect the harmonic component in series 2.

(iii) P(λ) test

This is a double window smoothing method, in which we choose a Bartlett window with n = 128, and a truncated periodogram window with m = 25. The P(λ) test statistic J_q is given in equation (8), and

$$\Lambda_{n,m} \approx 2n/3-2m+2m^2/n = 45.1.$$

Figure 4 plots P*(λ) vs. frequency where $p^*(\lambda) = p(\lambda)/C^*(0)$. Since J_q is a cumulative function of asymptotically normal distribution, the significant test can be constructed by plotting J_q vs. q and determining whether J_q crosses the boundary z_α where z_α can be derived from the usual two-sided percentage points of a standard normal. Thus, if α = 0.1, 0.05 and 0.01, we get $z_{.1}$ = 1.645, $z_{.05}$ = 1.96, and $z_{.01}$ = 2.58. The results are summerized in Table 4 which show that p(λ) test compared with Whittle's test is more powerful when the mass spectrum $f^y(\omega_k)$ is mixed with peaks of the spectral density function $f^z(\lambda_j)$, but is not as reliable when the harmonic component is separated from the peaks of the spectral density function of the noise.

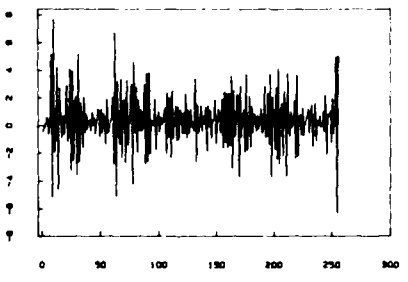
(iv) $G^{(b)}$ test process

To use the $G^{(b)}$ test, we first need to estimate the real part of the bispectrum $f^x(\lambda_j,0)$ at the frequency $(\lambda_j,0)$ by properly selecting the truncated lag and the two-dimensional smoothing window. Here we select the two-dimensional Bartlett window to be the smoothing function with M = 15.

Since, under H_0, $G^{(b)}$ has the same asymptotic distribution as $G^{(f)}$, we get $z_{.1}$ = .0559, $z_{.05}$ = .0615, $z_{.01}$ = .0743. Figure 5 shows the results of $I_N^x(\lambda_j)/Rf_N^x(\lambda_j,0)$ vs. frequency. The results are summarized in Table 5, which show that $G^{(b)}$ test actually detects the harmonic processes at the right frequency in both cases.

(v) $G^{(t)}$ test

Since the linear process $\{Z_t\}$ is generated from an exponential distribution, its fourth order cumulant certainly exists. We choose the 3-dimensional Bartlett window with lag M=10. Figure 6 shows the plot of $I_N^x(\lambda_j)/Rf_N^x(\lambda_j,0,0)$ vs. frequency. Table 6 presents the $G^{(t)}$ value for suspect peaks. The results show that the $G^{(t)}$ test detects the harmonic component at the right frequency in both cases also.



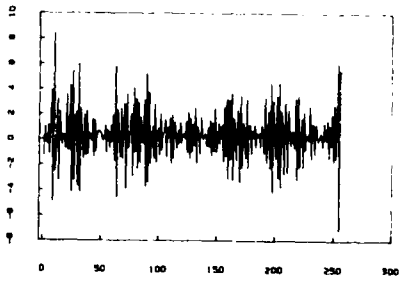


Figure 1. Time series plot for X_t

Figure 2. The periodogram plot for X_t

Figure 3. Plot of $I_N^X(\lambda_j)/f_N^X(\lambda_j)$

Figure 4. $P^*(\lambda)$ plot

Figure 5. Plot of $I_N^X(\lambda_j)/f_N^X(\lambda_j,0)$

Figure 6. Plot of $I_N^X(\lambda_j)/f_N^X(\lambda_j,0,0)$

SERIES #	$\Sigma \frac{I^x_N(\lambda)}{\lvert f^x_N(\lambda)\rvert}$	ω	$\frac{I^x_N(\lambda)}{\lvert f^x_N(\lambda)\rvert}$	$G^{(w)}$
1	118.1291	1.5708	7.1551	0.0606**
2	115.1999	2.6507	5.2138	0.0453

Table 1 Whittle's test

α level	4	8	12
0.1	.9077	.6398	.4778
0.05	.9267	.6737	.5037
0.01	.9571	.7407	.5764

Table 2 significant level for grouped periodogram test

SERIES #	K	$G^{(q)}_K(\omega_1)$
	4	0.9631***
1	8	0.8019***
	12	0.8431***
	4	0.7710
2	8	0.5382
	12	0.3344

Table 3 grouped periodogram test

SERIES #	θ	$P^*(\theta)$	$J_q(\theta)$
1	1.5708	0.1246	0.2317
2	2.6507	0.6364	2.0162**

Table 4 Priestley $P(\lambda)$ test

SERIES #	$\Sigma \frac{I^x_N(\lambda)}{\lvert f^x_N(\lambda,0)\rvert}$	ω	$\frac{I^x_N(\lambda)}{\lvert f^x_N(\lambda,0)\rvert}$	$G^{(b)}$
1	853.3934	1.5708	123.2088	.1444***
2	708.4138	2.6507	63.4764	.0896***

Table 5 $G^{(b)}$ test

SERIES #	$\Sigma \frac{I^x_N(\lambda)}{\lvert f^x_N(\lambda,0,0)\rvert}$	ω	$\frac{I^x_N(\lambda)}{\lvert f^x_N(\lambda,0,0)\rvert}$	$G^{(t)}$
1	6193.372	1.5708	1458.640	.2355***
2	4414.245	2.6507	498.3887	.1129***

Table 6 $G^{(t)}$ test

ACKNOWLEDGEMENTS. This research was supported in part by ONR contract N00014-85-K-0468.

REFERENCES

Bartlett, M. S. (1957), Discussion on "Symposium on spectral approach to time series." J.R. Stat. Soc. B, **19** , 1-63.

Brillinger, D. R. (1975), Time Series: Data Analysis and Theory. Holt, Rinehart and Winston, New York.

Brillinger, D. R., Rosenblatt, M. (1967), Asymptotic theory of k-th order spectra. In "Spectral Analysis of Time Series". (Ed. B. Harris), 153-188.

Fisher, R. A. (1929), Tests of significance in harmonic analysis. Proc. Roy. A, **125** , 54-59.

Lii, K. S., Rosenblatt, M. (1988) Bias and covariances of cumulant spectral estimates. Technical Report.

Priestley, M. B. (1962a) The analysis of stationary processes with mixed spectra-I. J.R. Stat. Soc. B, **24** , 215-233.

Priestley, M. B. (1962b) The analysis of stationary processes with mixed spectra-II. J.R. Stat. Soc. B, **24** , 511-529.

Priestley, M. B. (1981), Spectral Analysis and Time Series. Vols. I and II. Academic Press, New York.

Rosenblatt, M., Van Ness, J. W. (1965) Estimation of the bispectrum. Ann. Math. Stat., **36** , 1120-1136.

Schuster, A. (1898) On the investigation of hidden periodicities with application to a supposed 26-day period of metorological phenomena. Terr. Mag. Atmos. Elect., **3** , 13-41.

Siegel, A. F. (1980) Testing the periodicity in a time series. J. Amer. Stat. Ass., **75** , 345-348.

Whittle, P. (1952) The simultaneous estimation of a time series harmonic components and covariance structure. Trabajos. Estadist., **3** 43-57.

Whittle, P. (1954) The statistical analysis of a Seiche record. J. Marine Research., **13** , 76-100.

TIME SERIES IN A MICROCOMPUTER ENVIRONMENT

John D. Henstridge
Perth, Western Australia

ABSTRACT

The task of transfering a moderately large statistical package onto a microcomputer is described. It is shown that substantial consideration has to be given to the limitations of the microcomputer architecture but once this has been done the package can be made surprisingly efficient. The user interface also requires major adaptation so that it is more compatible with non-statistical microcomputer software. This leads to the necessity for interactive graphics and screen based operations.

Keywords: *microcomputers, statistical computing, time series.*

Introduction

TSA was first released in 1982 as a specialist time series package. It was was desiged with mainframe computers in mind and its strength lay in its ability to manipulate time series data. It was equally at home in both the time and frequency domains and used as its basic data types *series, filters* and *Fourier transforms*.

Early in 1987 it was decided to review the future of TSA both in terms of its statistical facilities and its implementation on various machines. A market analysis and feedback from users indicated that TSA needed strengthening in time domain model fitting (and forecasting) and that a personal computer (PC) version was needed to complement the mainframe versions. After a feasibility study it was decided to update TSA and have the PC version as the *primary version*. It was thought that the constraints of the PC were the greatest that were likely to be encountered and hence a PC version could be ported to a mainframe more readily than the reverse.

This paper describes what this involves and can be considered as a case study for the problem of transfering statistical software to the PC.

The Existing form of TSA

TSA Release 1 (Henstridge, 1980, 1982) was a program consisting of approximately 14000 lines of highly portable Fortran 66. It had its own command language similar to some structured structured dialects of BASIC, specially adapted to time series data. The series data type was different from vectors in many other packages in that a series had both a length and a starting time. Operations which would not make sense with vectors (such as adding together two vectors of different length) were properly defined for series in TSA. In addition Fourier transforms could be manipulated, with the TRANSFORM command could convert between series and transforms.

Some of the flavour of TSA can be gained from the following examples of TSA input which demonstrate two different methods of estimating the spectrum of a series X. The first approach is to fit a parametric model and then display the theoretical gain function of this:

```
QAIC 20 X %X
```

Fit an autoregressive model to X, using the Akaike criterion and store this as the filter %X

```
GAIN SQUARED %X
```

Display the square of the gain function of this filter.

The second approach is to form the Fourier transform and thus obtain the smoothed periodogram:

```
TRANSFORM X
```

Form the Fourier transform and names it ^X

```
SPECTRUM ^X
```

Display the spectrum by smoothing the periodogram formed from ^X

The flexibility of TSA derived from the fact that the data objects %X and ^X in the examples could be manipulated in themselves. For example the transform ^X could have arithmetic operations performed on it by the CALCULATE command or the filter %X could be used in the FILTER command to define new filters or to filter series.

The time domain model fitting facilities in TSA Release 1 emphasised univariate Box-Jenkins or ARIMA type models. The PRELIMINARY command would obtain preliminary estimates and the ARIMA command would then obtain least squares estimates. Separate commands such as QAIC above allowed for the quick fitting of autoregressive models, using automatic order selection if desired. The implementation of these commands used the NAG Library extensively. Its flexibility came from the way that models were stored as filters and could be operated on by all the filter commands in TSA.

Time Series Model Fitting

Time series modeling has a number of unique features which in combination distinguish it from other statistical model selection and fitting problems. These include

(i) The parameters are constrained to lie within a not readily described region.

(ii) The estimation procedure itself is non-linear and frequently has problems of local multicolinearity.

(iii) The structured index set (time itself) means that it is rarely possible to exclude parts of the data or to model subsets of the data separately.

(iv) There are many different possible diagnostic methods which could be used in model selection.

Methods are available which can automatically select models (Akaike, 1969 and many subsequent papers). These methods tend to be computationally very demanding and the theoretical results are almost exclusively asymptotic, with the small sample situation not being well understood except in the case of simple autoregressions. Consequently it is more common for a statistician to select a model by examining summary statistics such as the autocorrelation function followed by fitting several models and obtaining diagnostic statistics for these.

There are a variety of summary and diagnostic statistics in common use. The most commonly used are the autocorrelation function and the partial autocorrelation function, but use can also be made of spectra and various residual plots.

Needs of a Personal Computer Implementation

It is commonly thought that a program has to be cut down to fit onto a PC, but it is easy to foreget that the memory available on a PC is comparable to that available to the average interactive user on many mainframes only six or seven years ago. The PC is small compared with other machines today but any interactive statistical program which ran on a mainframe several years ago should be able to fit onto a PC today.

Instead the main constraints on the PC are

(i) The operating system is relatively crude and is supported by few good utilities. It cannot even be assumed that a user has access to a screen editor let alone a good graphics program. Hence a statistical program has to supply many features provided by the operating system on other computers. As a secondary issue for the software developer, until recently the Fortran compilers for the PC have been of doubtfull quality. Even now, PC Fortran compilers are not totally reliable.

(ii) The disk drives are relatively slow compared with mainframes, due partly to the operating system using only primitive algorithms to optimise the use of the disks. Today the speed of the disks tends to be a greater constraint than the CPU speed. This imposes serious constraints on overlaying a program – basically all overlay changes within a loop or iteration must be avoided.

(iii) The typical PC user has come to expect software which makes use of the wide communications bandwidth between the CPU and the screen. Windows, screen graphics, prompts and on-screen editing are all part of this. This is a good influence in the long term but it does create extra work for the software developer.

(iv) The typical PC user is not *paper oriented.* This is partly due to many users never knowing a batch environment where printed output was the only feedback from the computer but it is also related to the screen presentation being more dynamic than a printed page can ever be. It does mean that a statistical package must enable the user to display virtually any information at any time rather than assuming that the user has a printed copy of past output.

Implementation

The initial feasibility study for the PC version involved a direct port of the 32-bit mainframe version onto a PC. This was not a particularly difficult task since TSA had originally been written with highly centralised input and output modules and portability was a major design consideration. The initial PC version had about 360 Kbytes of code but was overlayed to run in less than 200 Kbytes. It made no use of special PC features and it has to be said that it ran slowly. The code size was surprising since it was somewhat larger than the code size on other computers (especially the VAX and a 68000 based Unix computer). The compiler being used

(Microsoft) was generally thought to produce efficient code so the causes for this were examined before proceding further.

Study of the code produced by the compiler made it clear that it was no use pretending that the PC is a 32 bit machine. The 8086 microprocessor has 16 bit registers and although it can address up to 1 Mbyte of memory, it is only efficient when the code and data are locally restricted to 64Kbyte segments. In addition the microprocessor has relatively few registers and most have various constraints on their use. These produced a substatial overheads on accessing large arrays and in parameter passing in subroutine calls. It became clear that major savings in code size were attainable by using 16 bit integers, modifying the addressing of COMMON blocks, reducing subroutine parameter lists and transfering the text of error messages to a separate file. The result was a 40% reduction in code size, and the program running several times faster. These changes in themselves required remarkably little change to the source code of TSA.

Program Enhancements

The savings in code size permitted a large number of enhancements to be made to the original version of TSA. In addition to a number of statistical additions detailed below these included extended graphics with output to the screen, printer or plotter, fuller control of input and output and an on-line help system.

The language of the package itself was extended in a number of ways to emphasise the time series application. Time leads and lags can now be applied to series in any situation using a postfix notation, and commands have been extended so that most common operations now require fewer commands. As an example of what this allows, the instruction

```
GRAPH X ON X(-1)
```

plots the values of the series X against the values for the previous time point. Many of these language changes did not involve increases in code size at all; instead they involved moving existing code from individual commands into the parser where they could be used by all commands.

The new graphics module extends the earlier concept of a *graphical layout* which is a TSA data structure giving all the parameters for a graphical display. All commands have a default layout, but this can be modified by the user and if required stored as a user defined layout. The guiding principle here has been to make appropriately labelled and scaled graphical displays readily accessable while not reducing the flexibility available to the user who needs it.

In addition a screen interface was developed. This uses a status line to divided the screen into an input window and an output window. Previous lines of input can be viewed and edited, while previous output can be scrolled back for viewing. The design of this screen interface was dictated by statistical considerations outlined below.

The final version consists of about 30000 lines of code, of which 97% is portable Fortran 77, 2% is Fortran specific to the PC and 1% is assembly language. On the PC it compiles into about 360 Kbytes and runs in about 370Kbytes of RAM. The program is overlayed only to a limited extent; overlay changes are sufficiently infrequent that it is feasible to run on a PC without a hard disk.

Special Facilities for Model Selection and Fitting

Most of the statistical additions to TSA were designed to aid in the selection and fitting of time domain models. They include transfer function modelling with options of least squares, maximum likelihood and marginal likelihood criteria (again using the NAG Library routines), the display of impulse response functions and a DECOMPOSE command which can use a variety of methods to decompose series into trend, seasonal and irregular components. These affect model fitting as follows:

(i) Selection is aided by the new DECOMPOSE command which can separate the seasonal, trend and irregular components. This is seen primarily as an aid to identification of models.

(ii) Details of all models fitted to a series are automatically stored. These details include the structure of the model, the parameter estimates, goodness of fit measure and the method of fitting (preliminary, least squares, maximum likelihood or marginal likelihood). It is possible to display these at any stage. The fact that the display of more than one model might not fit on the screen at one time was the major reason for the scrolling facility on the output window (since as outlined above, the user has probably not got a printed record).

(iii) A brief comparative display of all models fitted is also available. Together these two features provide the user with a means of managing a number of competing models.

(iv) Residual and fitted value series are automatically extracted for any model properly fitted. In addition the individual components of a model can be extracted as filters. These series and filters are then immediately available for diagnostic analysis by a number of TSA commands as shown in Figure 1.

(v) In the selection process it is common to fit models which are modifications on previously fitted models. Rather than implement a special command to edit an existing model, the input window editor was developed so that the input line where the previous model was specified could be edited and then entered as new input. This was considered to be visually easier and it gives a facility which can be used in other situations as well. In addition the filters which are extracted from previous models can be used to define components of a new model.

Summary

The final result of this work was a new version of TSA which had greatly improved functionality while still fitting on a standard PC. The most striking result was that the original portability of the program of the program could be maintained while giving a true PC implementation which makes use of the unique features of the PC. Experience has shown that a conversion to a different machine is less than a days work and it has been possible to automate most of this conversion process.

Most of the lessons learnt from this would apply to other statistical packages which are command oriented.

Aknowledgement
The author wishes to thank NAG Ltd., Oxford for the use of its facilities and access to the NAG Library for this work.

References

AKAIKE,H. (1969), Fitting autoregressions for predictions, *Annals of the Institute of Statistical Mathematics*, Tokyo, *21*, 243-247.

HENSTRIDGE, J. D., (1980), TSA – a package for time series analysis, *COMPSTAT-80*, ed. M.M. Barritt and D Wishart, Physica Verlag, Wien.

HENSTRIDGE, J. D. (1982), *TSA – an interactive package for time series analysis*, NAG, Oxford.

MOVING WINDOW DETECTION FOR 0-1 MARKOV TRIALS

Joseph Glaz, Philip C. Hormel and Bruce McK. Johnson*
University of Connecticut and CIBA-GEIGY Corporation

Abstract. Let $X_1, X_2, \ldots$ be a sequence of 0-1 Markov trials. The random variable X_i represents the number of signals that were detected at the end of the ith discrete-time interval. The k-out-of-m moving window detector generates a pulse whenever k or more signals are detected within m consecutive discrete-time intervals. Define $M_{k,m}$ to be the waiting time for detection using a k-out-of-m moving window detector. In this article we derive Bonferroni-type inequalities and product-type approximations for the distribution of $M_{k,m}$, which in turn yield approximations for $E(M_{k,m})$ and $Var(M_{k,m})$. These quantities play an important role in the design and analysis of the k-out-of-m moving window detection procedure. Applications to the theory of radar detection and quality control (zone tests) are discussed.

1. *Introduction.* Let $X_1, \ldots, X_n, \ldots$ be a sequence of 0-1 valued random variables. The random variable X_i represents the number of signals that were detected at the end of the ith discrete-time interval. The k-out-of-m moving window detector generates a pulse whenever k or more signals are detected within m consecutive discrete-time intervals. Define $M_{k,m}$ to be the waiting time for detection using a k-out-of-m moving window detector. Then $E(M_{k,m})$ and $Var(M_{k,m})$ are the mean and the variance of recurrence time, respectively, for at least k events in a moving window of size m. The waiting time, the mean and the variance of recurrence for detection using a k-out-of-m moving window detection procedure play an important role in a variety of applications. We describe below applications to quality control and radar detection.

In quality control, the sequence of 1's correspond to defective items. Greenberg [9], Roberts [20], and Saperstein [23] study the properties of the zone tests. They define the process to be "out of control" if a moving window of size m contains at least k observations outside a specified zone (say, the three sigma limits about the mean). The random variable $M_{k,m}$ is the waiting time between times at which the process is declared to be "out of control" and $1/E(M_{k,m})$ is the probability of type I error for testing the hypothesis that the process is "in control." For $k = m$ the zone is based on the run statistic, which is a special case of the moving window statistics, was introduced by Mosteller [15].

Consider a radar sweep where a dichotomous quantizer transmits to the detector the digit 1 if the signal-plus-noise waveform exceeds a predetermined threshold, and the digit 0 otherwise. Thus the data from a radar sweep is transformed into a random sequence of 0 and 1. The k-out-of-m moving window detector generates a pulse whenever k or more 1's were observed within a consecutive string of m elements. The moving window detection procedure has been discussed extensively in the literature ([1], [3], [5], [6], [14], [18], [25]). Dinneen and Reed [3] discuss signal detection and location by various digital methods. They conclude that "the moving window detector satisfies the detection and beam splitting criteria and at the same time is logically the simplest." Moreover, one can obtain a good estimate of the target center by employing the center of the window where at least k signals were observed. Bogush [1], Galati and Studer [5], Lefferts [14], Nelson [18] and Todd [25] study the moving window detection procedure under the assumption that the observed sequence of 0 and 1 is generated by a simple Markov process. The random variable $M_{k,m}$ is the waiting time between the times that the k-out-of-m moving window detector generates a pulse. The quantities $P(M_{k,m} > n)$, $E(M_{k,m})$ and $Var(M_{k,m})$ play an important role in the design of the moving window detector.

The evaluation of the quantities $P(M_{k,m} > n)$, $E(M_{k,m})$ and $Var(M_{k,m})$ is a formidable task. Even in the simpler situation, when the sequence of 0 and 1 are i.i.d. Bernoulli trials, these quantities can be evaluated only for limited values of k and m ([9], [12], [16] and [23]). Moreover, the methods developed for evaluating the quantities mentioned above in the i.i.d. Bernoulli case cannot be extended for the Markov model. Naus [17] and Samuel-Cahn [22] developed accurate approximations for the i.i.d. Bernoulli case, that too cannot be easily extended to the Markov model. For the Markov model, Glaz [6, Section III] derived a product-type lower bound for $P(M_{k,m} > n)$ and $E(M_{k,m})$. Since in many instances this lower bound is quite conservative (see Tables I-II in Section 3), there is a need for more accurate approximations.

In Section 2 we derive a product-type approximation for $P(M_{k,m} > n)$ that is far more accurate than the lower bound that was derived in [5]. This in turn yields accurate approximations for $E(M_{k,m})$ and $Var(M_{k,m})$. Recently, Hoover [11] derived Bonferroni-type upper (lower) bounds for a union (intersection) of a given sequence of events. For the problem at hand we evaluate in Section 2 the Bonferroni-type lower bounds for $P(M_{k,m} > n)$.

In Section 3, Tables I-II, we present the results of a simulation study for evaluating $P(M_{k,m} > n)$, $E(M_{k,m})$ and $Var(M_{k,m})$. These results are used to compare the product-type and Bonferroni-type approximations in Tables I-II. A discussion of the approximations is presented at the end of Section 3.

2. Product-type and Bonferroni-type approximations. Suppose that the observations $X_1, X_2, \ldots$ form a 0-1 sequence of Markov trials. Assume that

$$p = P(X_1 = 1)$$

and

$$p_i = P(X_j = 1 | X_{j-1} = i), \quad i = 0,1; \quad j = 2,3,\ldots.$$

We assume that the correlation coefficient of two successive observations is positive and $p = P(X_j = 1)$, $j \geq 1$. Therefore $X_1, X_2, \ldots$ is a two-state stationary homogeneous Markov chain and all the conditional probabilities can be derived from p and p_i, $i = 0,1$. Moreover, it follows that $p_0 < p < p_1$. The waiting time for the detection using a k-out-of-m window detector satisfies

$$M_{k,m} = \inf\{n \geq 1; \sum_{i=\max(1,n-m+1)}^{n} X_i \geq k\}, \quad (2.1)$$

which we will abbreviate to M. For $n \geq m \geq 2$ we define

$$\gamma_{n-m+1,n} = P\{\bigcap_{j=1}^{n-m+1} (\sum_{i=j}^{m+j-1} X_i < k)\}, \quad (2.2)$$

$$S(j,n) = \sum_{i=0}^{\min(j,n-j)} \binom{n-j}{j} \binom{j}{i} p_1^{j-i} (1-p_0)^{n-j-i} \cdot (p_0-p_1)^i, \quad (2.3)$$

and

$$S_k(n) = \sum_{j=0}^{k} S(j,n), \quad (2.4)$$

in terms of which we have the following results.

Theorem 2.1. Let M be the waiting time for detection, then for $k \leq n \leq m+2$

$$P(M > n) = \gamma_{\max(1,n-m+1),n},$$

where

$$\gamma_{1,n} = S_{k-1}(n-1) + (p_0-p_1)S_{k-2}(n-2) - pS(k-1,n-1), \quad k < n < m, \quad (2.5)$$

$$\gamma_{2,m+1} = \gamma_{1,m} - (1-p) \sum_{i=\max(0,2k-m-1)}^{k-1} \left[\binom{k-1}{k-i-1}\binom{m-k}{k-i-1} p_0^{k-i} p_1^i \cdot (1-p_0)^{m+1-2k+i} (1-p_1)^{k-i-1}\right], \quad (2.6)$$

and

$$\gamma_{3,m+2} = \gamma_{2,m+1} - (1-p) \sum_{i=\max(0,2k-m-2)}^{k-2} \left[\binom{k-2}{k-i-2}\binom{m-k}{k-i-2} p_0^{k-i-1} p_1^i \cdot (1-p_0)^{m+2-2k+i} (1-p_1)^{k-i-2}\right] \cdot \left[p_0(1-p_1) + \frac{m-2k+i+2}{k-i-1} p_1\right]. \quad (2.7)$$

Proof. To derive equation (2.5), we use the following results from Helgert [10]:

$$P(\sum_{i=1}^{n} X_i = j | X_1 = 1) = S(j-1,n-1) + (p_0-p_1)S(j-1,n-2)$$

and

$$P(\sum_{i=1}^{n} X_i = j | X_1 = 0) = S(j,n-1) + (p_0-p_1)S(j-1,n-2).$$

Therefore,

$$\gamma_{1,n} = P(\sum_{i=1}^{n} X_i < k) = \sum_{j=0}^{k-1} \binom{n}{j} p[S(j-1,n-1) + (p_0-p_1)S(j-1,n-2)] + (1-p)[S(j,n-1) + (p_0-p_1)S(j-1,n-2)].$$

Simplifying the expression given above results in equation (2.5).

To evaluate $\gamma_{2,m+1}$, note that

$$\gamma_{2,m+1} = \gamma_{1,m} - P(A),$$

where $A = (X_1=0, \sum_{i=2}^{m} X_i = k-1, X_{m+1}=1)$. The event A involves m transitions from the initial state 0 to the final state 1. Denote by $j_1 \to j_2$ the transition from state j_1 to state j_2, $j_1, j_2 = 0,1$. Let i be the number of $1 \to 1$ transitions. Then, $\max(0,2k-m-1) \leq i \leq k-1$. Since we have a total of k visits to state 1, we must have k-i $0 \to 1$ transitions. As the initial state is 0 and the final state is 1, the number of $1 \to 0$ transitions is equal to k - i - 1 (the number of $0 \to 1$ transitions minus one). Therefore, the remaining m + 1 + i - 2k transitions are of the type $0 \to 0$. Since there are $\binom{k-1}{k-i-1}$ ways of arranging the k - i - 1 $0 \to 1$ transitions between the first and the last 1 and $\binom{m-k}{k-i-1}$ ways of arranging the k - i - 1 $1 \to 0$ transitions between the first and the last 0, we get that

$$P(A) = (1-p) \sum_{i=\max(0,2k-m-1)}^{k-1} \binom{k-1}{k-i-1} \cdot \binom{m-k}{k-i-1} p_0^{k-i} p_1^i (1-p_0)^{m+1-i-2k} (1-p_1)^{k-i-1}.$$

Equation (2.6) follows.

To evaluate $\gamma_{3,m+1}$, define the events

$$A_1 = (X_1=0, X_2=0, \sum_{i=3}^{m} X_i = k-1, X_{m+1}=0, X_{m+2}=1),$$

$$A_2 = (X_1=0, X_2=0, \sum_{i=3}^{m} X_i = k-2, X_{m+1}=1, X_{m+2}=1),$$

and

$$A_3 = (X_1=1, X_2=0, \sum_{i=3}^{m} X_i = k-2, X_{m+1}=1, X_{m+2}=1).$$

Then it follows that

$$\gamma_{3,m+2} = \gamma_{2,m+1} - \sum_{j=1}^{3} P(A_j).$$

Using a similar enumeration technique for evaluating P(A) above, one obtains:

$$P(A_1) = (1-p) \sum_{i=\max(0,2k-m-2)}^{k-2} \binom{k-2}{k-i-2} \cdot \binom{m-k}{k-i-1} p_0^{k-i} p_1^{i} (1-p_0)^{m+2-2k+i} \cdot (1-p_1)^{k-i-1},$$

$$P(A_2) = (1-p) \sum_{i=\max(0,2k-m-2)}^{k-2} \binom{k-2}{k-i-2} \cdot \binom{m-k}{k-i-2} p_0^{k-i-1} p_1^{i+1} (1-p_0)^{m+3-2k+i} \cdot (1-p_1)^{k-i-2},$$

and

$$P(A_3) = p \sum_{i=\max(0,2k-m-2)}^{k-2} \binom{k-2}{k-i-2} \cdot \binom{m-k}{k-i-2} p_0^{k-i-1} p_1^{i+1} (1-p_0)^{m+2-2k+i} \cdot (1-p_1)^{k-i-1}.$$

Simplifying the expressions for $\sum_{j=1}^{3} P(A_j)$ yields equation (2.7). This concludes the proof of Theorem 2.1.

We now proceed to derive an approximation for $P(M > n)$. Let E_j denote the event $\sum_{i=j}^{j+m-1} X_i < k$, $j = 1,2,\ldots$. Then for $1 \le L \le n - m$

$$P(M > n) = P\left(\bigcap_{j=1}^{n-m+1} E_j\right) = P\left(\bigcap_{j=1}^{L} E_j\right) \prod_{j=L+1}^{n-m+1} P\left(E_j \mid \bigcap_{i=1}^{j-1} E_i\right) = \gamma_{L,m+L-1} \prod_{j=L+1}^{n-m+1} (\gamma_{j,m+j-1}/\gamma_{j-1,m+j-2}). \quad (2.8)$$

We propose to employ the following (L-1)th order "Markov-like" approximation for $P(E_j \mid \bigcap_{i=1}^{j-1} E_i)$:

$$P\left(E_j \mid \bigcap_{i=1}^{j-1} E_i\right) \approx P\left(E_j \mid \bigcap_{i=j-L+1}^{j-1} E_i\right), \quad j \ge L+1. \quad (2.9)$$

Substitute the right-hand side of equation (2.9) into equation (2.8) and use the stationarity of the events E_j, to get the desired product-type approximation:

$$P(M > n) \approx \gamma_{L,m+L-1} (\gamma_L^*)^{n-m-L+1} \equiv \pi_L, \quad (2.10)$$

where

$$\gamma_L^* = \gamma_{L,m+L-1}/\gamma_{L-1,m+L-2},$$

and $\gamma_{j,m+j-1}$ is defined in equation (2.2). The following result supports the approximation (2.10).

Theorem 2.2. Let M be the waiting time for detection given by equation (2.1). Then there exists a real number $0 < \gamma < 1$ such that

$$\lim_{n\to\infty} P(M > n+1 \mid M > n) = \gamma.$$

Proof. Let S be the set of all possible binary sequences of length m. Then the cardinality of S is 2^m. Out of all binary sequences with the property that their sum is greater or equal to k we create a single absorbing state. Let S* be the set containing the absorbing state and the remaining elements of the set S. Then $\{X_j + \ldots + X_{m+j-1}\}_{j=1}^{\infty}$ is a finite Markov chain with a single absorbing state and one set of communicating transient states. It now follows from Darroch and Seneta [2,§4] that there exists a constant $0 < \gamma < 1$ such that

$$\lim_{n\to\infty} P(M > n+1 \mid M > n) = \gamma.$$

This concludes the proof of Theorem 2.2.

It follows from Theorem 2.2 that the sequence of conditional probabilities $\gamma_{j,m+j-1}/\gamma_{j-1,m+j-2}$ become stationary as j increases. Therefore, it seems plausible to replace the terms $\gamma_{j,m+j-1}/\gamma_{j-1,m+j-2}$ for $j \ge L+1$ in equation (2.8) with $\gamma_L^* = \gamma_{L,m+L-1}/\gamma_{L-1,m+L-2}$. This substitution results in approximation π_L given by equation (2.10). In Section 3, Table I, we present numerical results for this approximation, in the case of L = 3.

We now turn to the problem of deriving Bonferroni-type lower bounds for $P(M > n)$. Recently Hoover (1987) has derived a sequence of Bonferroni-type upper bounds of order L, $1 \le L \le n-1$:

$$P\left(\bigcup_{i=1}^{n} A_i\right) \le P\left(\bigcup_{i=1}^{L} A_i\right) + \sum_{i=L+1}^{n} \sum_{\substack{i_j \in S_i \\ 1 \le i_1 < \ldots < i_L \le n}} P\left(A_i \cap \left[\bigcap_{j=1}^{L} \left(\bigcup A_{i_j}\right)^c\right]\right), \quad (2.11)$$

where $A_1,\ldots,A_n$ is a sequence of events, c denotes the complement of an event, and S_i is a subset of $\{1,\ldots,i-1\}$ of size L-1. For L = 1 and L = 2 the right-hand side in (2.11) reduces to the usual Bonferroni upper bound and Hunter Bonferroni-type upper bound (Hoover, 1987). In the case that $A_1,\ldots,A_n$ are naturally ordered in such a way that $P(\bigcap_{j=1}^{m} A_{i_j})$ is maximized for $i_j - i_{j-1} = 1$,

$2 \le j \le m \le n-1$, the natural ordering with $S_i = \{i-1,\ldots,i-L\}$ is recommended. In this case the upper bound in equation (2.11) reduces to:

$$P(\bigcup_{i=1}^{n} A_i) \le \sum_{i=1}^{n} P(A_i) - \sum_{i=1}^{n-1} P(A_i \cap A_{i+1}) - \sum_{j=2}^{L-1} \sum_{i=1}^{n-j} P\{A_i \cap (\bigcap_{t=1}^{j-1} A_{i+t}^c) \cap A_{i+j}\}.$$

If the events $A_1,\ldots,A_n$ are stationary, a further simplification of (2.11) is obtained:

$$P(\bigcup_{i=1}^{n} A_i) \le nP(A_1) - (n-1)P(A_1 \cap A_2) - \sum_{j=2}^{L-1} (n-j)P\{A_1 \cap (\bigcap_{i=1}^{j-1} A_{i+1}^c) \cap A_{j+1}\}. \tag{2.12}$$

The following result gives the Bonferroni-type lower bounds for $P(M > n)$.

Theorem 2.3. Let M be the waiting time for detection given by equation (2.1). Then for $L \ge 1$

$$P(M > n) \ge (n-m+2-L)\gamma_{L,m+L-1} - (n-m+1-L)\gamma_{L-1,m+L-2} \equiv \beta_L \tag{2.13}$$

where $\gamma_{0,m-2} \equiv 1$, and for $L \ge 1$ $\gamma_{L,m+L-1}$ are defined in equation (2.2). Moreover, for $L \ge 1$

$$\beta_{L+1} - \beta_L = (n-m+1-L)(\gamma_{L-1,m+L-2} - 2\gamma_{L,m+L-1} + \gamma_{L+1,m+L}). \tag{2.14}$$

Proof. It follows from equation (2.8) and (2.13) that for $L \ge 1$

$$P(M > n) \ge 1 - (n-m+1)P(E_1^c) - (n-m)P(E_1^c \cap E_2^c) - \sum_{j=2}^{L-1} (n-m+1-s)P\{E_1^c \cap (\bigcap_{i=1}^{j-1} E_{i+1}) \cap E_{j+1}^c\}, \tag{2.15}$$

where $E_j = (\sum_{i=j}^{j+m-1} X_i < k)$. It is routine to verify that the right-hand side of equation (2.15) simplifies to β_L, given by the right-hand side of equation (2.13). Equation (2.14) follows immediately from the definition of β_L. This completes the proof of Theorem 2.3.

In Section 3, Table I, we evaluate the performance of the lower bound β_L, for $L = 3$.

We now proceed to derive approximations for E(M) and Var(M), the mean and the variance of recurrence time, respectively, for at least k events in a moving window of size m. It is well-known, [4, 264-266], that

$$E(M) = \sum_{n=0}^{\infty} P(M > n)$$

and

$$Var(M) = 2\sum_{n=1}^{\infty} nP(M > n) + E(M)[1 - E(M)].$$

Note that for $n \le k-1$, $P(M > n) = 1$. Therefore,

$$E(M) = (k-1) + \sum_{n=k}^{m+L-1} \gamma_{\max(1,n-m+1),n} + \sum_{n=m+L}^{\infty} \gamma_{n-m+1,n} \tag{2.16}$$

and

$$Var(M) = (k-1)k + 2\sum_{n=k}^{m+1-1} nP(M > n) + 2\sum_{n=m+L}^{\infty} n\gamma_{n-m+1,n}. \tag{2.17}$$

Substitute in equations (2.16) and (2.17) for $\gamma_{n-m+1,n}$ its product type approximation $\gamma_{L,m+L-1}(\gamma_L^*)^{n-m-L+1}$. Then evaluate the respective geometric series to get:

$$E(M) \approx (k-1) + \sum_{n=k}^{m+L-1} \gamma_{\max(1,n-m+1),n} + \gamma_{L,L+m-1}\gamma_L^*/(1-\gamma_L^*) \equiv \hat{E}_L, \tag{2.18}$$

$$Var(M) \approx k(k-1) + 2\sum_{n=k}^{m+L-1} n\gamma_{\max(1,n-m+1)n} + 2\gamma_{L,L+m-1}\gamma_L^*[(m+L)(1-\gamma_L^*) + \gamma_L^*]/(1-\gamma_L^*)^2 + \hat{E}_L[1 - \hat{E}_L] \equiv \hat{V}_L, \tag{2.19}$$

where $\gamma_{j,j+m-1}$ are defined in equation (2.2) and $\gamma_L^* = \gamma_{L,m+L-1}/\gamma_{L-1,m+L-2}$. In Section 3, Table II, we evaluate the approximations $\hat{E}_L$ and $\widehat{SD}_L = \sqrt{\hat{V}_L}$, for $L = 3$.

Remark. The Bonferroni-type inequalities for $P(M > n)$ are not suitable for evaluating E(M) and Var(M) (for large n_0, $\beta_L < 0$ for $n \ge n_0$).

3. Numerical Examples. We now evaluate for selected values of m, k, p, p_1 and n the bounds and the approximations for $P(M > n)$, E(M) and $SD(M) = \sqrt{Var(M)}$, that have been derived in Section 2. These results are compared in Tables I-II with the approximations that have been derived in Glaz (1983).

For the simulated values of $P(M > n)$, E(M) and SD(M) (denoted in the tables below by SIM) 10,000 replicates of $N \approx 12 \times \widehat{SD}_3(M)$ pseudo-random numbers uniformly distributed on the interval (0,1) were generated, using IMSL routine GGUBS. Each of the uniform pseudo-random numbers were converted to an observation from a desired 0-1 Markov process. The reason for having to generate a sequence of Markov

TABLE I

APPROXIMATIONS FOR THE PROBABILITY OF THE WAITING TIME FOR DETECTION: TWO-STATE MARKOV CHAIN

m	k	p	p_1	n	LB π_1	LB β_3	π_3	SIM
10	3	.05	.10	20	.9148	.9474	.9479	.9503
				50	.5467	.8405	.8497	.8549
			.50	20	.8542	.8641	.8661	.8642
				50	.5985	.6482	.6858	.6819
		.10	.20	20	.6740	.7562	.7652	.7719
				50	.1407	.2992	.4567	.4715
			.80	20	.7270	.7266	.7324	.7373
				50	.4543	.3725	.4773	.4745
10	5	.05	.10	20	.9972	.9989	.9989	.9982
				50	.9897	.9962	.9962	.9974
			.50	20	.9558	.9645	.9647	.9634
				50	.8114	.8997	.9031	.9010
		.10	.20	20	.9558	.9787	.9788	.9780
				50	.8114	.9313	.9331	.9341
			.80	20	.8248	.8321	.8346	.8368
				50	.5855	.5872	.6361	.6341
25	3	.05	.10	35	.7416	.7539	.7679	.7700
				50	.5467	.6146	.6441	.6603
			.50	35	.7244	.7299	.7328	.7312
				50	.5985	.6080	.6291	.6268
		.10	.20	35	.3362	.3430	.3691	.3755
				50	.1407	.0368	.2040	.2262
			.80	35	.5767	.5756	.5806	.5680
				50	.4543	.4287	.4646	.4567
25	5	.05	.10	35	.9632	.9746	.9747	.9766
				50	.8858	.9532	.9538	.9593
			.50	35	.8915	.8993	.8998	.9038
				50	.8114	.8435	.8475	.8547
		10	.20	35	.7184	.7689	.7737	.7847
				50	.4491	.6040	.6390	.6687
			.80	35	.6991	.7042	.7069	.7065
				50	.5855	.5874	.6075	.6070

TABLE II

APPROXIMATIONS FOR THE EXPECTED AND STANDARD DEVIATION OF THE WAITING TIME FOR DETECTION: TWO-STATE MARKOV CHAIN

m	k	p	p_1	LB $\hat{E}_1(M)$	$\hat{E}_3(M)$	SIM E(M)	$\widehat{SD}_3(M)$	SIM SD(M)
10	3	.05	.10	68.46	280.16	299.10	274.36	294.37
			.25	65.06	184.80	188.11	180.65	185.03
			.50	77.10	130.56	130.19	128.53	128.91
		.10	.20	31.12	63.06	66.06	58.17	62.91
			.50	37.20	55.59	56.27	53.26	53.77
			.80	61.50	68.88	68.46	69.94	69.61
	5	.05	.10	3987.28	11232.33	11381.59	11224.67	11463.61
			.25	458.40	1894.20	1921.95	1887.91	1932.93
			.50	117.10	458.80	464.60	454.64	473.99
		.10	.20	194.30	634.17	654.03	627.10	656.02
			.50	57.20	176.54	175.87	172.03	172.50
			.80	81.50	111.07	109.93	110.35	108.88
25	3	.05	.10	61.06	98.08	108.88	85.48	98.19
			.25	65.06	97.96	104.29	88.36	95.93
			.50	77.10	103.12	105.93	98.47	102.34
		.10	.20	31.12	35.11	36.71	25.81	28.82
			.50	37.20	42.44	42.95	37.40	38.53
			.80	61.50	66.36	64.64	67.13	65.82
	5	.05	.10	114.78	708.03	826.66	690.23	791.41
			.25	105.06	413.94	464.56	399.21	456.29
			.50	117.10	259.18	272.58	250.21	267.06
		.10	.20	51.12	93.70	109.45	78.56	97.14
			.50	57.20	87.16	93.99	77.41	85.57
			.80	81.50	100.18	100.04	98.80	98.57

trials of length $N \approx 12 \times \widehat{SD}_3(M)$ is that the distribution of M has a very heavy right tail. The quantity $N \approx 12 \times \widehat{SD}_3(M)$ has been adopted after some numerical experimentation with evaluating E(M) via a simulation.

From the numerical results in Tables I-II, we can conclude that the new product-type approximations for $P(M > n)$ and E(M), given by π_3 and $\hat{E}_3(M)$, respectively, significantly improve the approximations π_1 and $\hat{E}_1(M)$ for these quantities that have been studied in Glaz (1983). The new approximations for SD(M), $\widehat{SD}_3(M)$ are also much more accurate than the approximation $\widehat{SD}_1(M)$. Moreover, the product-type approximation π_3 is more accurate than the Bonferroni-type lower bound β_3. In some cases the improvement of π_3 over β_3 is remarkable. For example, if m = 25, k = 3, p = .10, p_1 = .20 and n = 50, then β_3 = .0368, π_3 = .2040 and the simulated value of $P(M > 50)$ = .2262. Another deficiency of the Bonferroni-type lower bounds for $P(M > n)$ is that for $n \geq n_0$, it has a negative value. For this reason we have not evaluated the Bonferroni-type lower bound for E(M) and the related approximation for SD(M).

Although the new approximations provide us with quite accurate results in most cases, there is still room for improvement. For example, if m = 25, k = 5, p = .05 and p_1 = .10, the simulater values for E(M) and SD(M) are 826.66 and 791.41, respectively, while $\hat{E}_3(M)$ = 708.03 and $\widehat{SD}_3(M)$ = 690.23. This amounts to a relative error of 14% in approximating E(M) and 13% in approximating SD(M). To improve these approximations one can try to evaluate the approximations π_L, $\hat{E}_L(M)$ and $\widehat{SD}_L(M)$ for L > 3. We will report these results in a subsequent article.

Acknowledgment. The research in this article was partially supported by the Research Foundation of the University of Connecticut.

References

1. Bogush, Jr., A.J. (1972). Correlated clutter and resultant properties of binary signals. IEEE Trans. Aerosp. Electron. Syst. 9: 208-13.
2. Darroch, J.N. & Seneta, E. (1965). On quasi-stationary distributions in absorbing finite Markov chains. J. Appl. Prob. 2: 88-100.
3. Dinneen, G.P. & Reed, I.S. (1956). An analysis of signal detection and location by digital methods. IRE Trans. Inform. Theor. 2: 29-39.
4. Feller, W. (1970). An Introduction to Theory and Its Applications. Third Edition, Revised Printing. New York: John Wiley & Sons.
5. Galati, G. & Studer, F.A. (1982). Angular accuracy of the binary moving window detector. IEEE Trans. Aerosp. Electron. Syst. 18: 416-22.
6. Glaz, J. (1983). Moving window detection for discrete data. IEEE Trans. Inform. Theor. 29: 457-62.
7. Glaz, J. (1987). A comparison of Bonferroni-type and product-type inequalities in presence of dependence. Proc. Symp. on Dependence and Stat. and Prob, Somerset, Pensylvania, 1987.
8. Glaz, J. & Johnson, B. McK. (1984). Probability inequalities for multivariate distributions with dependence structures. J. Amer. Stat. Assoc. 79: 436-41.
9. Greenberg, I. (1970). The first occurrence of n successes in N trials. Technometrics 21: 627-34.
10. Helgert, H.J. (1970). On sums of random variables defined on a two-state Markov chain. J. Appl. Prob. 7: 761-65.
11. Hoover, D.R. (1987). Component complement addition upper bounds--an improved inclusion/exclusion method. ASA 1987 Proc. Stat. Comput. Section (in press).
12. Huntington, R.J. (1976). Mean recurrence times for k successes within m trials. J. Appl. Prob. 13: 604-07.
13. Karlin, S. & Ost, F. (1987). Counts of long alligned word matches among random letter sequences. Adv. Appl. Prob. 19: 293-351.
14. Lefferts, R.E. (1981). Adaptive false alarm regulation in double threshold radar detection. IEEE Trans. Aerosp. Electron. Syst. 17: 666-75.
15. Mosteller, F. (1941). Note on application of runs to quality control charts. Ann. Math. Stat. 12: 228-32.
16. Naus, J.I. (1974). Probabilities for a generalized birthday problem. J. Amer. Stat. Assoc. 69: 810-15.
17. Naus, J.I. (1982). Approximations for the distributions of scan statistics. J. Amer. Stat. Assoc. 77: 177-83.
18. Nelson, J.B. (1978). Minimal order models for false alarm calculations on sliding windows. IEEE Trans. Aerosp. Electron. Syst. 15:352-63.
19. Philippou, A.N. & Makri, F.S. (1986). Successes, runs and longest runs. Stat. & Prob. Lett. 4: 211-15.
20. Roberts, S.W. (1957). Properties of control chart zone tests. Bell Syst. Tech. J. 37: 83-105.
21. Samarova, S.S. (1981). On the length of the longest head-run for a Markov chain with two states. Theor. Prob. Appl. 26: 498-509.
22. Samuel-Cahn, E. (1983). Simple approximations to the expected waiting time for a cluster of any given size for point processes. Adv. Appl. Prob. 15: 21-38.
23. Saperstein, B. (1973). On the occurrence of n successes within N Bernoulli trials. Technometrics 15: 809-18.
24. Schwager, S.J. (1983). Run probabilities in sequences of Markov-dependent trials. J. Amer. Stat. Assoc. 78: 168-75.
25. Todd, P.H. (1981). Direct minimal-order Markov model for sliding-window detection probabilities. IEE Proc. 128: 152-54.

*Regrettably, Professor Johnson died on November 4, 1986.

Inference Techniques for a Class of Exponential Time Series

V. Chandrasekar
Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

P. J. Brockwell
Department of Statistics
University of Melbourne, Australia

Abstract: This research has been motivated by the need to study meteorological radar signals. The power received by a meteorological radar is the energy backscattered from an ensemble of meteorological targets. The time variation of this power can be modelled as a time series with exponential marginal distribution. Moreover the signals are observed at two polarization states of the transmitted wave and are correlated. This paper deals with the inference problem associated with the above described radar signals. We discuss two different schemes, one based on second order moments and the other using the distribution functions. The simulation study of these two schemes show that they have similar performance and hence the simpler moment technique can be used with real time radar applications.

I. Introduction

The time series under consideration in this paper is collected by a meteorological radar which receives the signals backscattered from an ensemble of hydrometeors (particles like raindrops, hail, ice, etc.). These particles also have a size distribution and orientation distribution associated with them. Thus we have an ensemble of particles that are randomly positioned, randomly distributed in size shape and orientation and move randomly. Fluctuation of the received power is related to all the above distributions. The marginal distribution of the received power is exponential in nature. The medium when observed at different polarizations give different mean powers due to the anisotropy of the medium, but they still are correlated since the observations come from the same set of targets. Statistical properties of these dual polarized signals have been studied by Bringi, et al., 1983. Simultaneous observation of the targets at two polarizations is difficult to achieve technologically, and as a result the observation is made at two polarizations switching between them very fast. This creates an inherent property of missing observation.

Thus we have a class of multivariate exponential time series describing the backscattered power received by a meteorological radar. This paper deals with inference problem associated with this exponential time series and the paper is organized as follows: Section 2 deals with the description of the exponential series in terms of complex Gaussian time series. Section 3 analyzes the one step predictors namely moment method and conditional expectation. In section 4 we obtain the corresponding two step predictors and Section 5 presents the conclusions with a summary of key results of this paper.

2. The Exponential Time Series

Let $\underset{\sim}{x}_t = \underset{\sim}{U}_t + i\underset{\sim}{V}_t$, $t \in \{0, \pm 1, \ldots\}$ be the m-variate zero mean complex-valued stationary Gaussian series. If we restrict our attention to the case when $\{\underset{\sim}{U}_t\}$ and $\{\underset{\sim}{V}_t\}$ are uncorrelated sequences, then we can construct the exponential series as $\{\underset{\sim}{P}_t\}$ where $P_{jt} = U_{jt}^2 + V_{jt}^2$ where U_{jt}, V_{jt} and P_{jt} are the jth components of the real and imaginary parts. $\underset{\sim}{U}_t$, $\underset{\sim}{V}_t$ of the multivariate Gaussian series $\underset{\sim}{x}_t$. Properties of this series and the relationship with the complex Gaussian series have been studied by Chandrasekar et al (1987), and we refer to that paper for details.

In this paper we deal with constructing the likelihood functions for the exponential series. Let $\underset{\sim}{x}_t$ be an n-dimensional complex vector with mean vector c and positive definite covariance matrix R. That is

$$E(x) = c \quad \text{and} \tag{1}$$

$$E(x - c)(\bar{x} - \bar{c})' = R \quad \text{where}$$

$(\bar{x} - \bar{c})'$ indicates the transpose of complex conjugate. The quadratic form $(\bar{x} - \bar{c})' R^{-1} (x-c)$ is real. Then we can write the density function $f(x)$ as

$$f(x) = \frac{1}{\pi^n \det R} \exp\left[-(\bar{z}-\bar{c})' R^{-1}(z-c)\right] \tag{2}$$

where $f(x)$ is a real-valued scalar function of the complex vector x, see Miller, (1974). From the above density function we can get the density of P by integrating over the phases of the full complex vector.

The above model of the exponential series obtained from complex Gaussian fits the radar data very well. The power received by radar comes from the square of in-phase and quadrature component of the received signal that behaves complex Gaussian.

The spectra of radar signals are approximately Gaussian in nature. This implies that the autocorrelation function is also Gaussian. These autocorrelation functions can be written in terms of a spectrum width σ_v, sampling time of the radar T_s and wavelength of the signals λ. The autocorrelation function of the complex signal at lag m, $(\rho(m))$ can be written in terms of the above parameters as

$$\rho(m) = \exp\left[-8\left(\frac{\pi\sigma_v \cdot m\, T_s}{\lambda}\right)^2\right] e^{-j\,\frac{4\pi\,\bar{v}\,m\,T_s}{\lambda}} \qquad (3)$$

It can be shown that $\rho_p(m)$, the autocorrelation function at lag 'm' of the power signals is related to $\rho(m)$ as

$$\rho_p(m) = |\rho(m)|^2 \qquad (4)$$

3. Two Step Predictor

Let $\underset{\sim}{r}_t$ be amplitude vector that corresponds to the power vector $\underset{\sim}{P}_t$ where $P_{jt} = r_{jt}^2$. We can either predict in amplitude domain or power domain. The amplitude time series can easily be constructed as the term by term square root of the power series. We consider only univariate stationary time series for the sake of inference analysis and these results can be extended easily to multivariate cases with the introduction of appropriate cross correlation functions. We consider two predictors here for comparison, based on the second order moments and the density function as follows:

Predictor I: This predictor based on inner products is constructed as follows (Brockwell and Davis, 1987):

$$\hat{P}_{i+1} = a_o + a_1\ P_i \qquad (5)$$

where a_o and a_1 are evaluated based on the criteria that $\hat{P}_{i+1}$ is a projection of P_{i+1} on the space P_i with the constraint that $E(\hat{P}_{i+1}) = E(P_{i+1})$ to obtain an unbiased estimator. Under this condition we get $\hat{P}_{i+1}$ as

$$\hat{P}_{i+1} = (1 - |\rho|^2) + |\rho|^2(P_i) \qquad (6)$$

where ρ^2 is the lag 1 auto-correlation of the power time series, see Chandrasekar et al. (1987). The above result is valid for unit mean and variance and for realistic signals can be scaled accordingly.

Predictor II: This predictor is obtained taking the conditional expectation of r_{i+1} conditioned on r_i,

$$r_{i+1} = E[r_{i+1}|r_i] \qquad (7)$$

computation of the above result requires knowledge of the joint density function or the conditional expectation and is obtained as follows: Let

$$x_1 = r_1 \exp(i\theta_1) \quad \text{and}$$
$$x_2 = r_2 \exp(i\theta_2) \qquad (8)$$

where r_j, θ_j are defined as $r_j = |x_j|$ and θ_j the corresponding argument of x_j. This transformation enables us to work in the domain of amplitude and phase of the complex signal X. Let S be the inverse of the covariance matrix with its terms defined as

$$S = R^{-1} = \begin{bmatrix} S_{11} & S_{12} \\ \bar{S}_{12} & S_{22} \end{bmatrix} \qquad (9)$$

where s_{11}, s_{22} are positive real and s_{12} is complex. The complex density function now becomes

$$f'(r_1,r_2,\theta_1,\theta_2) = \frac{r_1 r_2 \det s}{\pi^2} \exp(-\bar{x}'\, s\, x) \qquad (10)$$

We can obtain the joint distribution of amplitudes by integrating over the phases as

$$f(r_1,r_2) = 4\, r_1 r_2 (\det s) \exp\left[-(s_{11} r_1^2 + s_{22}\, r_2^2)\right] \cdot I_o(2|s_{12}|r_1 r_2) \qquad (11)$$

$$r_1,\ r_2 \geq 0$$

$$= 0 \quad \text{elsewhere}$$

where I_o is the modified Bessel function of the first kind and order zero, see Miller (1974). Similarly it can be shown that the marginal distribution g(r) is given by

$$g(r) = \frac{2r}{\bar{p}} \exp\left(\frac{-r^2}{\bar{p}}\right) \qquad (12)$$

where $\bar{p}$ is the mean power of the signal. Using equations (11) and (12) we can write the conditional distribution $g(r_2|r_1)$ as

$$g(r_2|r_1) = r_2 \det(S) \exp\left[-(s_{11} r_1^2 \cdot r_1^2 + s_{22}\, r_2^2)\right] I_o(2|s_{12}|r_1 r_2) \qquad (13)$$

Integrating equation (11) over the range of r_2 (0 to ∞) we can obtain predictor 2.

The conditional density function for radar signals can be obtained from the autocorrelation description of Section 2 and (13) as

$$g(r_2/r_1) = \exp\left(\frac{-|\rho|^2 r_1^2}{1-|\rho|^2}\right)\left[\frac{2r_2}{(1 - |\rho|^2} \exp \cdot \left(\frac{-r_2^2}{1 - |\rho|^2}\right) \cdot I_o\left(\frac{2\rho}{1 - |\rho|^2}\ r_1\, r_2\right)\right] \qquad (14)$$

The conditional density given by (13) can be integrated multiplying by r_2 to get the value of predictor II.

Example 1: The conditional density $g(r_2|r_1)$ is integrated over the entire range of r_2(0 to ∞) to compute the conditional expectation and is calculated numerically. It would be useful to make some comments on this numerical computation. The typical measurements made by radars give a value of $|\rho|^2$ very close to unity and the term $1 - |\rho|^2$ has to be handled carefully when it appears in denomenators. Next the value of I_ν for large arguments increase exponentially and they have to be cancelled explicitly with other terms to make the expectation stable.

We use typical radar parameters to compute the predictor II and they are as follows

λ = 10 cm (Microwave radar at S band)
T_s = 1 millisecond.
Mean power of signal is unity.

Figure 1a shows the one step predictor, $\hat{p}_{i+1}$ (predictor II) as a function of the signal p_i with the spectrum width σ_v as parameter. The various curve markings x, +, Δ,... □ indicate values of σ_v = 1, 2,...6 respectively. We can see that for large values of spectrum width the predictor is nearly unperturbed by the value of p_i but for narrow spectra the predictor increases nearly linearly with p_i. This gives a suggestion that a linear predictor may do almost as good as predictor 2. Figure 1b shows predictor 1 for radar parameters identical to those used in predictor 2. This is a linear predictor and the results are only slightly higher for all values of p_i and all spectrum width. The above phenomena can be easily explained based on shape of the distributions. Predictor 2 uses the information on the distribution of the signals which falls exponentially with signal power and hence weighs it lower than the predictor 1 which does not make use of the shape of the distribution. However, the difference between the two estimators is relatively small.

Several simulated time series were used to test these two predictors. Mean square errors were calculated on the difference between the known signal p_{i+1} and $\hat{p}_{i+1}$ applying the two predictors on simulated series, see Chandrasekar et al., (1987). The difference between the mean square errors for the two predictors were small leading us to conclude that the two one step predictors perform similarly.

4. Two Step Predictor:

The two step predictor in principle is an extension over the one step predictor but gets complicated quickly. We again consider two predictors here similar to section 3 based on second order moments and density function.

Predictor I: This predictor is based on inner products and is constructed as follows:

$$\hat{P}_{i+2} = b_o + b_1 P_i + b_2 P_{i+1} \tag{15}$$

where b_o, b_1 and b_2 are evaluated based on the criteria that $\hat{P}_{i+2}$ is a projection of P_{i+2} on the space containing P_i and P_{i+1} with the constraint that $E(\hat{P}_{i+2}) = E(P_{i+2})$ to obtain our unbiased estimator. Under these conditions we get b_j, j = 0,1,2 as

$$b_0 = \frac{1 - |\rho(2)|^2}{1 + |\rho(1)|^2}$$

$$b_1 = \frac{|\rho(2)|^2 - |\rho(1)|^4}{1 - |\rho(1)|^4} \tag{16}$$

$$b_2 = \frac{|\rho(1)|^2[1 - |\rho(2)|^2}{[1 - |\rho(1)|^4}$$

The above result is valid for unit mean of the power signal and for non unit means the predictor can be scaled accordingly.

Predictor II: This predictor is obtained similar to the one step predictor as

$$\hat{r}_{i+2} = E[r_{i+2}|r_i, r_{i+1}] \tag{17}$$

$$\text{Let } x_j = r_j \exp(i\,\theta_j)\ j = 1,2,3 \tag{18}$$

Similar to those discussed in Section 3

$$S = R^{-1} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ \bar{S}_{12} & S_{22} & S_{23} \\ \bar{S}_{13} & \bar{S}_{23} & S_{33} \end{bmatrix}$$

Then we can write

$$f(r_1, r_2, r_3) = \int_0^{2\pi}\int\int \frac{\det s}{\pi^4}\ \bar{X}'\, S\, X\, d\theta_1\, d\theta_2\, d\theta_3 \tag{19}$$

Equation (19) can be reduced after some algebraic manipulations as, (Miller, 1974)

$$f(r_1, r_2, r_3) = 8(\det s)\, r_1 r_2 r_3 \cdot \exp[-(r_1^2 S_{11} + r_2^2 S_{22} + r_3^2 S_{33})]$$
$$\cdot \sum_{m=0}^{\infty} \epsilon_m (-1)^m I_m(2r_1 r_2 |S_{12}|) I_m(2r_2 r_3 |S_{23}|) \cdot$$
$$I_m(2r_3 r_1 \cdot |S_{31}|)\cos m(\phi_{12} + \phi_{23} + \phi_{31}) \tag{20}$$

where ϕ_{12}, ϕ_{23} and ϕ_{31} are phases of S_{12}, S_{23}, S_{31} respectively, ϵ_m = 1 for m = 0 and 2 for m = 1,2,..., and I_m is the modified Bessel function of the first kind and order m.

We can write predictor 2 as

$$\hat{r}_{i+2} = E\left[\frac{f(r_1, r_2, r_3)}{f(r_1, r_3)}\right] \qquad (21)$$

Equations (20) and (21) indicate the complexity involved in computation of $\hat{r}_{i+2}$. The integrant in computing the expectation is our infinite summation containing terms with modified Bessel function. It is simpler numerically to compute the three dimensional integration in Eq. (19) and then use it in (21) to compute the expectation. This integration of the complex density function turns out to be computationally intensive and more involved than one step predictor.

Example 2: This example is constructed for the same radar parameters as in example 1. Figure 2 shows the two step predictor $\hat{P}_{i+2}$ as a function of P_{i+1} for $P_i = 1$ and mean power of unity. The continuous curve shows the results of predictor 2 whereas the points indicate predictor 1. We can see from Fig. 2 that, though there is some difference between the two predictors for small values of P_{i+1}, the overall agreement seems good between the two predictors. We cannot make a conclusive statement based on this example; however we see the trend same as in example 1, (i-e) the two predictors perform similarly. This observation is important in the context of the computational complexity involved in obtaining predictor 2.

Conclusions and Discussion:

We have discussed some inference techniques for a class of exponential time series in the context of radar signals. The exponential time series is constructed from complex Gaussian time series with arbitrary correlation structure.

Two predictors have been considered for analysis, one based on second order moments (predictor I) and the second based on conditional expectation (predictor II). These two predictors have been derived for one step and two step prediction. The amount of complexity involved in higher order predictors for predictor 2 is exhibited clearly by a comparison of the corresponding one step and two step predictors. The moment method predictors are computationally simple compared to predictor II. The computational complexity of two step predictor II is an order of magnitude more than that of one Step Predicator II. Real time applications of predictor II will be possible only through a pre-calculated look up table system since these are computationally intensive. The smooth variation of these predictors as observed in examples 1 and 2 indicate that we may not need too many entires in the look up table and can possibly be interpolated.

We have done a mean square error criteria evaluation of these two predictors for one step prediction based on simulation and they both give nearly equal mean square errors. The example for two step predictor shows that, over a wide range on the average, the two predictors give similar values. In real time applications in a radar, simplicity of computation is as important as accuracy as long as we can keep making real-time updates of observational data. Thus based on the above observations and radar system constraints our initial suggestions is that moment based predictor I is suited well for radar applications. The time series studied here is reversible and hence all the results discussed here can easily be extended to other inference problems.

Acknowledgements

This research was supported by the Center for Geosciences at Colorado State University sponsored by the Army Research Office. (DALL03-86-K-0175)

References

Chandrasekar, V., P. J. Brockwell and V. N. Bringi, 1987: Simulation of multivariate exponential time series, 19th Symposium on the Interface, Computer Science and Statistics, ASA, Washington, D. C.

Miller, K. S., 1974: Complex stochastic processes, Addison-Wesley, London.

Brockwell, P. J. and R. A. Davis, 1987: Time Series: Theory and Applications. Springer Verlag, New York.

Bringi, V. N., T. A. Seliga, and S. M. Cherry, 1983: Statistical properties of the dual-polarization differential reflectivity (Z_{DR}) radar signal. IEEE Trans. Geo. Sci. Remote Sensing, 21, 215-220.

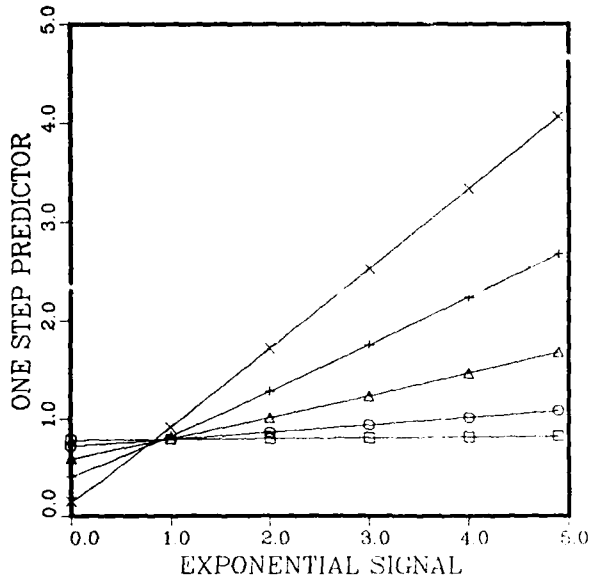


Figure 1a. Plot of $\hat{P}_{i+1}$ (predictor II) as a function of P_i with spectrum width σ_v as parameter. The curve markings x, +, Δ...□ indicate values of σ_v = 1,2,...6 respectively.

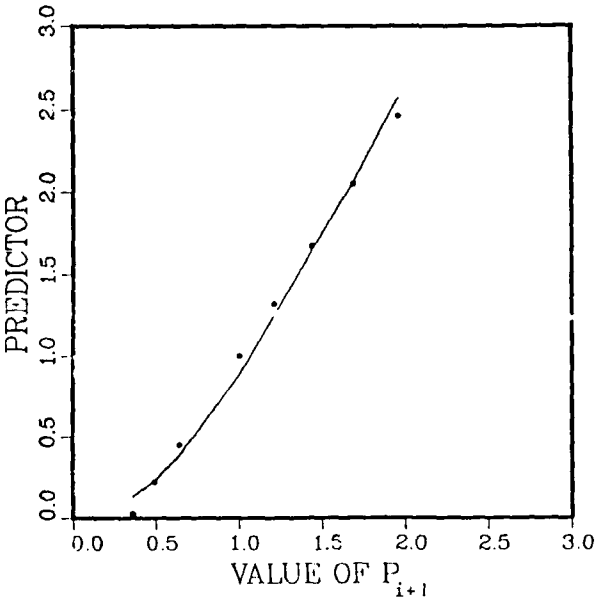


Figure 2. Plot of two step predictor ($\hat{P}_{i+2}$) as a function of P_{i+1} with P_i = 1 and unit mean signal. The continuous curve shows predictor 2 whereas the marked points indicate predictor 1.

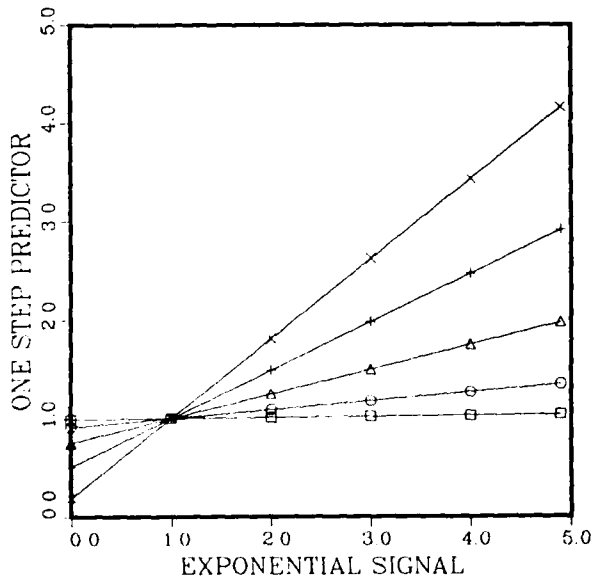


Figure 1b. Same as Figure 1a except that $\hat{P}_{i+1}$ is calculated as predictor I.

ALTERNATIVE METHODS FOR COMPUTING THE THEORETICAL AUTOCOVARIANCE FUNCTION OF MULTIVARIATE ARMA PROCESSES: A COMPARISON

Stefan Mittnik, SUNY at Stony Brook*

Abstract

Matricial expressions relating the theoretical autocovariances of multivariate autoregressive moving average processes to the parameters of the process are employed to derive a framework which unifies alternative algorithms for computing theoretical autocovariance functions.

1. INTRODUCTION

The theoretical autocovariance function of autoregressive moving average (ARMA) processes has to be computed frequently in applied time series analysis. For example, exact maximum likelihood estimation procedures for ARMA models require the derivation of theoretical autocovariances in each iteration of the maximization algorithm (see, for example, Nicholls and Hall, 1979; Gardner et al., 1980; Shea, 1987; Mittnik, 1988a). Theoretical autocovariances are also needed in distributional analyses of estimated ARMA parameters (Hannan, 1970) and for correctly initializing simulations with ARMA models (Ansley and Newbold, 1980; Woodfield, 1988).

The problem of computing the theoretical autocovariance function amounts to specifying and solving a system of linear equations. Algorithms for univariate ARMA processes have been suggested in McLeod (1975, 1977) Akaike (1978) and Tunnicliffe Wilson (1979). In the multivariate case the specification of the system's coefficient matrix represents a major difficulty. Nicholls and Hall (1979) present an algorithm for the multivariate processes. The algorithm given in Pate and Davies (1988) is essentially equivalent to the one of Nicholls and Hall. Ansley (1980) and Kohn and Ansley (1982) propose slightly more efficient algorithms by eliminating some of the unknowns in the equation system. By developing a closed form matricial relationship, which expresses theoretical autocovariances of multivariate ARMA processes in terms of the ARMA parameters, Mittnik (1988b) presents a procedure which, compared to the previous approaches, reduces the number of unknowns in the system by a factor of about two.

In this paper we employ the matricial expressions in Mittnik (1988b) to compare the alternative approaches to computing theoretical autocovariance functions of multivariate ARMA processes. While the earlier algorithms require rather complex indexing schemes to set up the coefficient matrices (see, for example, Ansley, 1980; Kohn and Ansley, 1982; Pate and Davies, 1988), the results derived here yield closed form expressions for the coefficient matrices of the respective algorithms.

2. ARMA COEFFICIENTS AND AUTOCOVARIANCES

Assume the stationary zero mean non-deterministic time series $\{y_t\}$, $y_t \in R^m$, is generated by the ARMA(p,q) process

$$A(L)y_t = B(L)\varepsilon_t, \qquad (2.1)$$

where A(L) is a stable matrix polynomial in the lag operator L defined by $A(L)=I-A_1L-A_2L^2-\ldots -A_pL^p$, and $B(L)=B_0+B_1L+\ldots+B_qL^q$. Process $\{\varepsilon_t\}$ is white noise, i.e. $E(\varepsilon_t)=0$ and $E(\varepsilon_s\varepsilon_t^\tau)=\delta_{st}\Sigma$. Note that without loss of generality either B_0 or Σ can be assumed to be an identity matrix. Unless stated otherwise we set $\Sigma=I$.

It is well known that given the initial autocovariances $\Gamma_\tau=E(y_t y_{t-\tau}^\tau)$ $(\tau=0,\ldots,p-1)$ of an ARMA(p,q) process, higher order autocovariances can be determined recursively applying the modified Yule-Walker equations. From the definition of the autocovariance it follows that

$$\Gamma_\tau = A_1\Gamma_{\tau-1}+\ldots+A_p\Gamma_{\tau-p}+E(B_0\varepsilon_t y_{t-\tau}^\tau+\ldots+B_q\varepsilon_{t-q}y_{t-\tau}^\tau) \quad (\tau=0,1,\ldots) \qquad (2.2)$$

* Department of Economics, SUNY at Stony Brook, Stony Brook, NY 11794-4384.

Replacing $y_{t-\tau}$ in (2.2) by its moving average representation, $y_{t-\tau}=A^{-1}(L)B(L)\varepsilon_{t-\tau}=C(L)\varepsilon_{t-\tau}$, where $C(L)=C_0+C_1L+\ldots$, and recalling the unit variance assumption, we can write

$$E(\varepsilon_{t-i}y_{t-\tau}^T) = \begin{cases} C_{i-\tau}^T, & i=\tau,\tau+1,\ldots,q \\ 0, & \text{otherwise.} \end{cases}$$

Defining $\Gamma=(\Gamma_0^T\ \Gamma_1^T\ \ldots\ \Gamma_p^T)^T$, $\Gamma^*=(\Gamma_0\ \Gamma_1\ \ldots\ \Gamma_p)^T$, $C=(C_0^T\ C_1^T\ \ldots\ C_q^T)^T$, $C^*=(C_0\ C_1\ \ldots\ C_q)^T$ and using the fact that $\Gamma_{\tau-i}=\Gamma_{i-\tau}^T$ allows us to rewrite (2.2) in matrix terms as (Mittnik, 1988b)

$$\Gamma = M_T\Gamma + M_H\Gamma^* + NC^* \qquad (2.3)$$

where the $m(p+1)\times m(p+1)$ matrices M_H and M_T are defined as

$$M_H = \begin{bmatrix} 0 & H \\ 0 & 0 \end{bmatrix}, \qquad M_T = \begin{bmatrix} 0 & 0 \\ T & 0 \end{bmatrix}.$$

Matrix H denotes the Hankel matrix

$$H = \begin{bmatrix} A_1 & A_2 & \ldots & A_{p-1} & A_p \\ A_2 & A_3 & \ldots & A_p & 0 \\ \vdots & & & & \vdots \\ A_{p-1} & A_p & & & \\ A_p & 0 & \ldots & \ldots & 0 \end{bmatrix}$$

and T is the lower triangular Toeplitz matrix

$$T = \begin{bmatrix} A_1 & 0 & \ldots & 0 \\ A_2 & A_1 & & \vdots \\ \vdots & & \ddots & 0 \\ A_p & A_{p-1} & \ldots & A_1 \end{bmatrix};$$

finally, the $m(p+1)\times m(q+1)$ matrix N is defined by

$$N = \begin{bmatrix} B_0 & B_1 & \ldots & B_{q-1} & B_q \\ B_1 & B_2 & \ldots & B_q & 0 \\ \vdots & & & & \vdots \\ B_{p-1} & B_p & & & \\ B_p & B_{p+1} & \ldots\ B_q & 0 & \ldots\ 0 \end{bmatrix}, \quad \text{if } p<q;$$

$$N = \left[\begin{array}{cccc} B_0 & B_1 & \ldots & B_q \\ B_1 & B_2 & \ldots & 0 \\ \vdots & & & \vdots \\ B_q & 0 & \ldots & 0 \\ \hline 0_{m(p-q)\times m(q+1)} & & & \end{array}\right], \quad \text{if } p\geq q.$$

Note that if $\Sigma\neq I$, matrix C^* is defined by $C^*=(C_0\Sigma\ \ldots\ C_q\Sigma)^T$. Expression (2.3) relates the theoretical autocovariances to the autoregressive coefficients and the coefficients of the pure moving average representation. A matricial expression purely in terms of the ARMA parameters can be found in Mittnik (1988b).

3. COMPARISON OF ALGORITHMS

Nicholls and Hall's (1979) procedure for computing the initial p+1 theoretical autocovariance matrices amounts to vectorizing the transposition of (2.3), yielding

$$\gamma = (M_T\otimes I_m)\gamma + (M_H\otimes I_m)W_{p+1}\gamma + \delta, \qquad (3.1)$$

where $\gamma=\mathrm{vec}(\Gamma^T)$, $\delta=\mathrm{vec}(C^{*T}N^T)$, and matrix $W_{p+1}=I_{p+1}\otimes W$ is a 0-1 commutation matrix such that $\mathrm{vec}(\Gamma^T)=W_{p+1}\mathrm{vec}(\Gamma^{*T})$ and $\mathrm{vec}(\Gamma_i^T)=W\mathrm{vec}(\Gamma_i)$. By defining

$$M = I - M_T\otimes I_m - (M_H\otimes I_m)W_{p+1}$$

we obtain the linear equation system

$$M\gamma = \delta, \qquad (3.2)$$

whose solution provides the elements of the initial p+1 autocovariance matrices. The approaches of Nicholls and Hall (1979) and Pate and Davies (1988) correspond to solving the $m^2(p+1)$-dimensional system (3.2).

Using the fact that Γ_0 is symmetric and extending the univariate results of McLeod (1975), Ansley (1980) reduces the size of the system by eliminating the $m(m-1)m/2$ redundant elements in Γ_0. Letting γ^1 denote the vector obtained by eliminating the redundant elements in γ, we can define the 0-1 matrices S_1 and S_2 such that $\gamma^1=S_1\gamma$, $\delta^1=S_1\delta$ and $\gamma=S_2\gamma^1$. The equation system providing the solution for γ^1 is

$$S_1 M S_2 \gamma^1 = \delta^1. \qquad (3.3)$$

Expression (3.3), involving $mp+(m+1)m/2$ unknowns, is equivalent to Ansley's (1980) approach. The coefficient matrix, which he constructs with a rather complex indexing scheme, corresponds to matrix $S_1 M S_2$ in (3.3).

Observing that

$$\Gamma_p = \sum_{i=0}^{p-1} A_{p-i}\Gamma_i + K_p, \qquad (3.4)$$

where

$$K_p = \begin{cases} \sum_{i=p}^{q} B_i C_{i-p}^T, & \text{if } p \le q \\ 0, & \text{if } p > q, \end{cases} \qquad (3.5)$$

and that Γ_p^T affects only Γ_0 in (2.3), Kohn and Ansley (1982) eliminate Γ_p from the equation system. In our framework, this is accomplished by substituting (3.4) for Γ_p in the RHS of (2.3). Defining $\tilde{\gamma}=\mathrm{vec}(\Gamma_0\ \Gamma_1^T\ \ldots\ \Gamma_{p-1}^T)$, we can write

$$\tilde{\gamma} = (\tilde{M}_T \otimes I)\tilde{\gamma} + (\tilde{M}_H \otimes I)W_p\tilde{\gamma} + M_p V\tilde{\gamma} + \tilde{\delta},$$

where $\tilde{M}_T$ and $\tilde{M}_H$ are obtained by deleting both the last block row and last block column of M_T and M_H, respectively,

$$\tilde{\delta} = \mathrm{vec}(K_p A_p^T + K_0^T \quad K_1^T \ \ldots\ K_{p-1}^T),$$

with K_i denoting the i^{th} (block) entry in NC^*,

$$M_p = (A_p^T \mid 0_{m \times m(p-1)})^T \otimes (A_p\ A_{p-1}\ \ldots\ A_1),$$

$W_p = I_p \otimes W$, and matrix V is defined such that $\mathrm{vec}(\Gamma_0^T \ldots \Gamma_{p-1}^T)^T = V\mathrm{vec}(\Gamma_0^T \ldots \Gamma_{p-1}^T)$. Let

$$\tilde{M} = I - \tilde{M}_T \otimes I + \tilde{M}_H \otimes I + M_p$$

and define γ^2 as the vector obtained by deleting the redundant elements in $\tilde{\gamma}$. Moreover, define the 0-1 matrices S_3 and S_4 such that $\gamma^2 = S_3\tilde{\gamma}$, $\delta^2 = S_3\tilde{\delta}$ and $\tilde{\gamma} = S_4\gamma^2$. The theoretical autocovariances $\Gamma_0, \ldots, \Gamma_{p-1}$ are calculated by solving

$$S_3 \tilde{M} S_4 \gamma^2 = \delta^2, \qquad (3.6)$$

a system with $m^2p - m(m-1)/2$ unknowns. Computing the autocovariances via (3.6) corresponds to the algorithm in Kohn and Ansley (1982).

Making use of the particular structure of (2.3), Mittnik (1988b) proposes a more efficient algorithm. Partitioning Γ such that $\Gamma = (\Gamma^{1T}\ \Gamma^{2T})^T$, with $\Gamma^1 = (\Gamma_0^T \ldots \Gamma_s^T)^T$, $\Gamma^2 = (\Gamma_{s+1}^T \ldots \Gamma_p^T)^T$, where

$$s = \begin{cases} \frac{p-2}{2}, & \text{if } p \text{ is even} \\ \frac{p-1}{2}, & \text{if } p \text{ is odd,} \end{cases}$$

enables us to rewrite (2.3) as

$$\begin{bmatrix} \Gamma^1 \\ \Gamma^2 \end{bmatrix} = \begin{bmatrix} T_{11} & 0 \\ T_{21} & T_{22} \end{bmatrix}\begin{bmatrix} \Gamma^1 \\ \Gamma^2 \end{bmatrix} + \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & 0 \end{bmatrix}\begin{bmatrix} \Gamma^{*1} \\ \Gamma^{*2} \end{bmatrix} + \begin{bmatrix} K^1 \\ K^2 \end{bmatrix}, \qquad (3.7)$$

where matrices T_{ij}, H_{ij}, and K^i denote conformable submatrices of M_T, M_H, and NC^*, respectively. By defining

$\tilde{\gamma}_1 = \mathrm{vec}(\Gamma^{1T}) = W_{s+1}\mathrm{vec}(\Gamma^{*1T})$ $\quad$ $\tilde{\delta}_1 = \mathrm{vec}(K^1)$

$\gamma_2 = \mathrm{vec}(\Gamma^{2T}) = W_{p-s}\mathrm{vec}(\Gamma^{*2T})$ $\quad$ $\delta_2 = \mathrm{vec}(K^2)$

$T_1 = \Gamma_{11} \otimes I$ $\quad$ $H_1 = (H_{11} \otimes I)W_{s+1}$

$T_2 = T_{21} \otimes I$ $\quad$ $H_2 = (H_{21} \otimes I)W_{p-s}$

$T_3 = T_{22} \otimes I$ $\quad$ $H_3 = (H_{12} \otimes I)W_{s+1}$,

we can write

$$(I - T_1 - H_1)\tilde{\gamma}_1 = H_3\gamma_2 + \tilde{\delta}_1 \qquad (3.8)$$

$$(I - T_3)\gamma_2 = (T_2 + H_2)\tilde{\gamma}_1 + \delta_2. \qquad (3.9)$$

Substituting

$$\gamma_2 = (I - T_3)^{-1}[(T_2 + H_2)\tilde{\gamma}_1 + \delta_2] \qquad (3.10)$$

into (3.8) and eliminating again the redundant elements in $\tilde{\gamma}_1$ by defining S_5 and S_6 such that $\gamma_1 = S_5\tilde{\gamma}_1$, $\delta_1 = S_5\tilde{\delta}_1$ and $\tilde{\gamma}_1 = S_6\gamma_1$, gives

$$S_5 M_1 S_6 \gamma_1 = S_5 H_3 (I - T_3)^{-1}\delta_2 + \delta_1, \qquad (3.11)$$

where

$$M_1 = I - T_1 - H_1 - H_3(I - T_3)^{-1}(T_2 + H_2).$$

Once γ_1 has been computed, γ_2 can be obtained either recursively (Mittnik, 1988b) or from

(3.10). The number of unknowns in system (3.12) is $(m^2p+m)/2$ if p is odd and $(m^2(p-1)+m)/2$ if p is even. Thus, for large values of p the number of unknowns in (3.12) is substantially less than for any of the other methods.

The ratio of the elementary multiplications of the (previously most efficient) algorithm by Kohn and Ansley (1982) over the ones required in Mittnik (1988b), reported in Table 1, indicates the computational savings of the latter approach. Note, the fact that the construction of the coefficient matrices in (3.11) are more complex has to be taken into consideration. Using, however, Akaike's (1973) block-Levinson algorithm, the inversion of of $I-T_3$ requires only $O(p^2)$ operations.

Table 1: Comparison of Computational Complexity

m	p	Ratio
1	2	7.0
	3	4.2
	4	6.0
	6	6.2
2	2	9.5
	3	3.5
	4	8.4
	6	8.2
3	2	12.7
	3	3.8
	4	9.7
	6	9.0
4	2	15.2
	3	4.1
	4	10.5
	6	9.5
5	2	17.0
	3	4.2
	4	11.0
	6	9.8

m: number of variables
p: autoregressive order
Ratio: ratio of elementary multiplications Kohn&Ansley (1982)/Mittnik (1988b)

4. CONCLUSIONS

Making use of a matricial expression relating the autocovariances of an ARMA process to the ARMA parameters, a general framework for alternative approaches to computing the theoretical autocovariance function of multivariate ARMA models has been provided. The results facilitate the implementation of these algorithms by deriving closed form expressions for their respective coefficient matrices instead of using complex indexing schemes.

REFERENCES

Akaike, H. (1973), Block Toeplitz Matrix Inversion, SIAM Journal of Applied Mathematics, 24, 234-241.

Akaike, H. (1978), Covariance Matrix Computation of the State Variable of a Stationary Gaussian Process, Annals of the Institute of Statistical Mathematics, 30, 499-504.

Ansley, C.F. (1980), Computation of the Theoretical Autocovariance Function for a Vector ARMA Process, Journal of Statistical Computation and Simulation, 12, 15-24.

Gardner, G., A.C. Harvey, and G.D.A. Phillips (1980), An Algorithm for Exact Maximum Likelihood Estimation of ARMA Models by Means of the Kalman Filter, Applied Statistics 29, 311-322.

Kohn, R. and C.F. Ansley (1982), A Note on Obtaining the Theoretical Auto-covariances of an ARMA Process, Journal of Statistical Computation and Simulation, 15, 273-283.

McLeod, A.I. (1975), Derivation of the Theoretical Autocovariance Function of Autoregressive-moving Average Processes, Applied Statistics, 24, 255-256.

McLeod, A.I. (1977), Correction, Applied Statistics, 26, 194.

Mittnik, S. (1987), Non-recursive Methods for Computing the Coefficients of the Autoregressive and the Moving-average Representation of Mixed ARMA Processes, Economics Letters, 23, 279-284.

Mittnik, S. (1988a), Exact Maximum Likelihood Estimation of Multivariate ARMA Models via Kalman Filtering, Dept. of Economics Working Paper, No. 304.

Mittnik, S. (1988b), Computation of Theoretical Autocovariance Matrices of Multivariate ARMA Time Series, unpublished manuscript.

Nicholls, D. and A.D. Hall (1979), The Exact Likelihood Function of Multivariate Autoregressive-moving average Models, Biometrika, 66, 259-264.

Pate, M.B. and N. Davies (1988) Computation of Population Correlation Matrices in MARMA(P,Q) Time Series, Applied Statistics, 37, 127-155.

Shea, B.L. (1987), Estimation of Multivariate Time Series, Journal of Time Series Analysis, 8, 95-109.

Tunnicliffe Wilson, G., Some Efficient Computational Procedures for High Order ARMA Models, Journal of Statistical Computation and Simulation, 8, 301-309.

Woodfield, T. (1988), Simulating Stationary Gaussian ARMA Time Series, this volume.

XVI. RELIABILITY AND LIFE DISTRIBUTIONS

INCREASING RELIABILITY OF MULTIVERSION FAULT-TOLERANT SOFTWARE DESIGN BY MODULATION

Junryo Miyashita,California State University at San Bernardino

Abstract:
One of the problems of the multi-version fault-tolerant software is the high cost of development. This paper addresses that problem . Rather than working on the common requirement specification for a whole program, teams of programmers will work on the common specifications for each module in each version of a program. One version of a program consists of a set of modules. This will enable the modules in each version to be interchangeable. The effects on the reliability by such modularization scheme are studied. Theoretical reliability of modularized N-version Programming and Recovery Block are derived in closed forms assumming independence among different modules. Numerical results show substantial increase in the reliability in both N-version Programming and Recovery Block schemes.

Section One : Introduction.

Fault-tolerance in software is achieved by introducing redundancy.There are three well known designs : N-version Programmings, Recovery Block, and Consensus Recovery Block. All three designs effect reliability by using multiple implementations of a common requirements specification.One problem with multiversions is the high development cost.This paper addresses the problem.

Reliability is increased without the increase in costs by delaying the introduction of redundancy untill later in the development cycle. Redundancy is introduced after completion of a modular design which provides a common set of module specificaitons. Multiple versions of each module are then implemented; as opposed to multiple versions of an entire program. Each version of module is interchageable with every other version. Multiversions of a program are built by assembling different version of the requisite modules. Theoretical studies provide further confidence in the efficacy of this approach. Theoretical reliabilities of modularized ,multiversion fault-tolerant software are derived in closed form assumming the independence among modules. The effects of modularization on reliability for the multiversion designs are calculated. The result show substantial increases in reliability for each.

Section Two : Effect of modularization of N-version programming.

Part One: General Formula.

We shall assume that we have N independent versions of a software and in each version, there are M independent modules.

In N-version programming, we shall assume that we have a correct output when two of the output agree. Now the effect of modularization is that as long as any two versions of different modules are correct then the two outcomes of those modules are assumed to be correct. In other word, the reliability of non-modulalized N-version programming is that Pr(at least two versions of each modules's outputs agree for all modules.).

We also note here that we can process all the possible permutations (ie. N-versions,M modules : NM permutaions) and add to each module the information as to which version it is.Then if there are agreements in output, we can check if agreements come from "independent" permutations : permutations which do not share any common version at any module level.

We shall define following terms.

R(i,j) = Reliability of i-th versions of j-th module.

R(i) = Reliability of i-th version

$$= \prod_{j=1}^{M} R(i,j)$$

Rnvp = Reliability of N-version programming without modularization.

RNVP-M = Reliability of N-version Programming with modularlization.

Then we have,

RNVP = Pr(At least two versions agree(correct))

$$= 1 - \left[\prod_{i=1}^{N} (1-R(i)) + \sum_{i=1}^{N} \left(R(i) \prod_{s<i} (1-R(s)) \right) \right] \quad (1)$$

RNVP-M = Pr(At least two versions are correct at each module level)

$$= \prod_{j=1}^{M} Pr(\text{ At least two versions are correct at j-th level }) = \prod_{j=1}^{M} \{ 1 - Pr(0 \text{ error}) - Pr(1 \text{ error}) \} = \prod_{j=1}^{M} \left(1 - \prod_{i=1}^{N} (1-R(i,j)) + \prod_{i=1}^{N} (R(i,j) * \prod_{s<i} (1-R(i,j))\right) \quad (2)$$

Part Two: Numerical Result on special case when reliability of each module are the same. (i.e. R(i,j) = R for all i and j).

If R(i,j) = R for all i and j then,

RNVP = 1 - [(1-RM)N-1 * (1+(N-1)*RM)]

RNVP-M = [1 - (1-R)N-1 * (1+(N-1)*R)]M

The three table of values of RNVP ,RNVP-M and the ratio $\frac{1-R_{NVP}}{1-R_{NVP-M}}$ is given at next pages (Table 1,2,and 3). The

last table is for the ratio of errors where the numerator is the probability of failure in N-version programming without modulation and the denominator is the probability of error with modulation. The results show the substantial increase in reliability. For example, when the reliability of each module is constantly .98 and 8 module with each having 5 versions ,will increase the reliability 327 times. We must remember here that we assummed independence among modules which may not likely to be true.

Section Three. Modularization effect on Recvovery Block Model.

Part One: Comments on general case.

In Recovery Block Model, We have additional factors to consider, namely

1) The reliability of acceptance test(s),
2) Recovery reliability .
3) Arbitrary ordering of versions to be excecuted.

In this paper, we shall have a simple assumption that acceptance test has perfect reliability as well as recovery. Then the ordering of versions does not affect the reliability of the entire scheme because the probability that at least one correcct version existing becomes the reliability of the Recovery Block scheme.

We shall define following terms.

RRB = Reliability of Recovery Block Model without modularization.

RRB-M = Reliability of Recovery Block Model with modularization.

Then we have,

RRB = Pr(At least one version is correct

$$= 1 - \prod_{i=1}^{N}(1 - R(i))$$

Here R(i) = Reliability of one version

$$= \prod_{j=1}^{M} R(i,j)$$

RRB-M = Pr(At least one version is correct for each module)

$$= \prod_{j=1}^{M} Pr(\text{ At least one version is correct at j-th module level })$$

$$= \prod_{j=1}^{M} (1 - Pr(\text{No correct version at j-th module level})).$$

$$= \prod_{j=1}^{M} (1 - \prod_{i=1}^{N} (1 - R(i,j)))$$

Now if we assume that R(i,j) = R for all i and j then ,

$$RRB = 1 - (1-RM)N$$

and

$$RRB\text{-}M = (1 - (1-R)N)M$$

Again, the table of R_{RB} ,R_{RB-M}, and $\frac{1 - R_{RB}}{1 - R_{RB-M}}$ are given (Table 4,5 and 6). The tables show the substantial increase in the reliability by modularization.

REFERENCES:

[1] Keith Scott , Data Domain Modeling of Fault-Tolerant Software Reliability, Ph.D. Dissertation, Dep[artment of Electrical and Computer Engineering , North Carolina State University , Raleigh, NOrth Carolina, 1983.

[2] T. Anderson and P.A. Lee, Fault Tolerance , Prentice/Hall International, 1981.

[3] R. Keith cott, James W Gault and David F. McAllister, "Modelling Fault-Tolerant Software Reliability," Proceedings of the Third Symposium on Reliability in Distributed Software and DataBase System. 1983.

[4] H. Hecht " Fault-Tolerant Software, " IEEE Transactions on Reliability , Vol. R-28, No. 3, August 1979,pp. 227-232.

[5] Aligirdas Avizienis and Liming Cheng."On the Implementationof N-version Programming for Software Fault-Tolerance During Program Execution," Proceedings of COMPSAC 1977, pp 149-155.

[6] George J. Schick and Ray W. Wolverton,"An Analysis of Cometing Software Reliability Model" IEEE Transactions on Software Engineering, Vol. SE-4, No 2, March 1978.

[7] C. V. Ramamoorthy and F.B. Bastani, "Software Reliability -- Status and Perspectives, " IEEE transactions on Software Engineering, Vol. SE-8, No 4, July 1982 , pp 354-371c

r = reliability of one module in one version
N = number of independent versions
M = number of module per version

r = .94

N\M	2	3	4	5	6	7	8
2	.780749	.6898698	.609569	.5386151	.4759203	.4205232	.371574
3	.9625073	.9236198	.8768662	.8252617	.7711148	.7161698	.661722
4	.9942423	.9830212	.964774	.939675	.9084379	.8720594	.831647
5	.9991676	.9964392	.9904725	.9802682	.9652221	.9451244	.920105

r = .96

N\M	2	3	4	5	6	7	8
2	.8493465	.7827577	.7213896	.6648326	.6127097	.5646732	.520402
3	.9825241	.9632053	.9387492	.9103251	.8789226	.8453752	.810382
4	.9981858	.994404	.9878682	.978312	.9656716	.9500289	.931569
5	.999823	.9991988	.9977348	.9950484	.9907989	.9847116	.976587

r = .98

N\M	2	3	4	5	6	7	8
2	.9223682	.8858424	.8507631	.817073	.784717	.7536421	.723798
3	.9954198	.9900316	.9828557	.9740802	.9638796	.9524142	.939832
4	.9997589	.9992223	.9982375	.9967079	.9945587	.9917337	.988193
5	.9999881	.999943	.9998296	.9996067	.9992284	.9986473	.997815

Table 1: Reliability of non-modulated version
when reliability of each module is r and
n = number of version and m = number of modules

r = .94

N\M	2	3	4	5	6	7	8
2	.780749	.6898698	.609569	.5386151	.4759203	.4205232	.3715743
3	.9793715	.9692174	.9591685	.9492238	.9393822	.9296428	.9200042
4	.9983503	.9975265	.9967034	.9958809	.9950591	.9942381	.9934176
5	.9998766	.9998149	.9997532	.9996916	.9996299	.9995682	.9995066

r = .96

N\M	2	3	4	5	6	7	8
2	.8493465	.7827578	.7213896	.6648326	.6127097	.5646733	.5204029
3	.9906777	.9860492	.9814423	.9768569	.972293	.9677504	.9632291
4	.9995034	.9992552	.9990071	.9987591	.998511	.9982631	.9980152
5	.9999752	.9999628	.9999504	.999938	.9999256	.9999132	.9999008

r = .98

N\M	2	3	4	5	6	7	8
2	.9223682	.8858425	.8507631	.817073	.7847169	.7536421	.723798
3	.9976334	.9964522	.9952725	.9940941	.9929172	.9917416	.9905674
4	.9999369	.9999054	.9998739	.9998424	.9998108	.9997793	.9997478
5	.9999983	.9999975	.9999967	.9999958	.999995	.9999942	.9999933

Table 2: The reliabilities of modulated versions.

r = .94

N\M	2	3	4	5	6	7	8
2	1	1	1	1	1	1	1
3	1.817518	2.481277	3.015656	3.441344	3.775878	4.034128	4.22868
4	3.490191	6.864355	10.68551	14.64541	18.53157	22.20449	25.5763
5	6.74686	19.23961	38.60991	63.96986	93.95765	127.0928	161.904

r = .96

N\M	2	3	4	5	6	7	8
2	1	1	1	1	1	1	1
3	1.874644	2.637466	3.300566	3.874808	4.369925	4.794632	5.15672
4	3.653583	7.513806	12.21863	17.47666	23.05496	28.7696	34.4768
5	7.139423	21.54167	45.67789	79.87885	123.6923	176.1648	236.052

r = .98

N\M	2	3	4	5	6	7	8
2	1	1	1	1	1	1	1
3	1.935422	2.809798	3.626523	4.388822	5.099723	5.762057	6.37866
4	3.822306	8.221173	13.97448	20.88166	28.76182	37.45234	46.8071
5	7.142857	22.76191	51.03572	94.27143	154.1191	231.5714	327.160

Table 3:
This table shows the ratio of probabilities of errors
The denominator is the probability of error when modulation is introduced and the numerator is when no modulation is introduced

r = .94

N\M	2	3	4	5	6	7	8
2	.986451	.9712982	.951929	.9291929	.9038192	.8764319	.847563
3	.9984229	.9951374	.9894604	.9811585	.9701714	.9565631	.940484
4	.9998164	.9991762	.9976891	.9949864	.9907492	.9847309	.976763
5	.9999787	.9998604	.9994934	.9986659	.9971311	.9946326	.990927

r = .96

N\M	2	3	4	5	6	7	8
2	.9938534	.9867142	.9773035	.9659127	.9528058	.9382216	.922376
3	.9995181	.9984686	.9965208	.9937066	.9897474	.9846448	.978373
4	.9999622	.9998234	.9994849	.9988381	.9977727	.9961834	.993974
5	.999997	.9999797	.9999224	.9997854	.9995161	.9990513	.998321

r = .98

N\M	2	3	4	5	6	7	8
2	.9984318	.9965416	.9939733	.9907688	.9869681	.9826091	.977728
3	.9999379	.9997966	.9995321	.9991131	.9985122	.9977066	.996676
4	.9999975	.999988	.9999636	.9999148	.9998301	.9996976	.999504
5	.9999999	.9999993	.9999971	.9999918	.9999806	.9999601	.999926

Table 4: The reliabilities of non-modulated Recovery Block

r = .94

N\M	2	3	4	5	6	7	8
2	.9928129	.9892388	.9856775	.9821291	.9785934	.9750704	.9715601
3	.9995681	.9993521	.9991362	.9989204	.9987047	.9984889	.9982732
4	.999974	.999961	.999948	.999935	.9999221	.9999091	.9998961
5	.9999983	.9999975	.9999967	.9999958	.999995	.9999942	.9999933

r = .96

N\M	2	3	4	5	6	7	8
2	.9968025	.9952076	.9936152	.9920254	.9904382	.9888534	.9872713
3	.999872	.9998079	.9997439	.9996799	.9996159	.9995519	.9994879
4	.9999949	.9999923	.9999898	.9999872	.9999846	.9999821	.9999795
5	.9999998	.9999996	.9999995	.9999994	.9999993	.9999992	.9999991

r = .98

N\M	2	3	4	5	6	7	8
2	.9992002	.9988004	.998401	.9980016	.9976024	.9972034	.9968046
3	.999984	.9999761	.' 99681	.9999601	.9999521	.9999441	.9999361
4	.9999996	.9999994	.9999993	.9999991	.9999989	.9999988	.9999986
5	1	1	1	1	1	1	1

Table 5: The reliabilities of modulated versions of Recovery Block.

r = .94

N\M	2	3	4	5	6	7	8
2	1.885187	2.667154	3.356325	3.962131	4.493034	4.956684	5.359952
3	3.651166	7.50506	12.20247	17.45296	23.0278	28.74531	34.4671
4	7.06422	21.13303	44.46101	77.16973	118.656	167.8716	223.5379
5	12.78571	55.76191	151.7857	319.7572	573.0119	918.8776	1359.018

r = .96

N\M	2	3	4	5	6	7	8
2	1.922304	2.772272	3.554809	4.274503	4.935713	5.542383	6.098332
3	3.763967	7.97393	13.35335	19.6622	26.69305	34.27139	42.23467
4	7.372093	22.96124	50.24419	90.66977	144.8372	212.7309	293.8663
5	12.5	56.83333	162.75	360	676.5	1136.857	1760.313

r = .98

N\M	2	3	4	5	6	7	8
2	1.960653	2.883081	3.769002	4.619382	5.435426	6.218457	6.969764
3	3.88806	8.487562	14.64552	22.20896	31.04478	41.02026	52.01866
4	7	22.33334	50.83333	95.33334	158.3333	241.6191	346.75
5	large	large	large	large	large	large	large

Table 6: The ratio of errors : The numerators is the probabilities of errors for non-modulated version and the denominators are for modulated version.

LINEAR PREDICTION OF FAILURE TIMES OF A REPAIRABLE SYSTEM

M. Ahsanullah, Rider College

1. ABSTRACT

Suppose we consider a repairable system in which a failed component is replaced immediately by a component of equal age. On replacement of the component, the system becomes operational and the repairing time of the component is assumed to be negligible. We assume the survival times of the components are independent and identically distributed. Some distributional properties of the n-th survival time are discussed when the survival times have different life distributions. Various predictions of the s-th failure time X_s $(s > n)$ based on the first n failure times are obtained.

2. INTRODUCTION

We consider a repairable system in which a failed component is replaced immediately by a component of equal age and the system becomes operational. Let us denote by $X_0, X_1, X_2, \ldots,$ the failure times of the system where $X_0 = 0$. The time between failures $U_n = X_n - X_{n-1}$, $n \geq 1$ are non negative random variables. Let $F(t) = P(U_1 \leq t)$, for $t \geq 0$ and $\bar{F}(t) = 1 - F(t)$.

We assume that $F(t)$ has a density $f(t)$ with $F(0) = 0$ and $r(t) = f(t)(\bar{F}(t))^{-1}$, for $\bar{F}(t) > 0$. The function $r(t)$ is called the hazard rate and $R(t) = \int_0^t r(u)du$ is called the cumulative hazard rate. Let $F_n(t) = Pr(X \leq t)$ and $f_n(t) = F_n'(t)$.

Then

$$1 - F_{(n)}(x) = \bar{F}(x) \qquad \text{if } n = 1$$

$$= \bar{F}(x) + \bar{F}(x)\, R(x) \qquad \text{if } n = 2$$

and in general

$$1 - F_{(n)}(x) = \bar{F}(x) \sum_{i=0}^{n-1} (R(x))^i\, (i!)^{-1}.$$

$1 - F_{(n)}(x)$ can be interpreted as the survival time to the n-th failure of the system given that the failed components of the system was replaced by one of equal age and the repair times were negligible. The probability density function (pdf) $f_n(x)$ of X_n can be written as

$$f_{(n)}(x) = f(x)\, \frac{(R(x))^{n-1}}{(n-1)!} \quad , \; n \geq 1, \; x \geq 0$$

$$= 0, \text{ otherwise.}$$

If F is the distribution function of a non negative random variable, we will call F is 'new better than used' (NBU), if $\bar{F}(x+y) \leq \bar{F}(x)\, \bar{F}(y)$ for all $x, y \geq 0$ and F is 'new worse than used' (NBU), if $\bar{F}(x+y) \geq \bar{F}(x)\, \bar{F}(y)$, for all $x, y \geq 0$. We will say F belongs to the class c_1 if F is either NBU or NWU. We will say F belongs to c_2 if $r(x) = f(x)(\bar{F}(x))^{-1}$, $\bar{F}(x) > 0$, is either monotone increasing or decreasing.

For various life distributions, the distributional properties of the n-th survival time are discussed. Linear prediction of the s-th failure time based on the first n $(n < s)$ failure time is given.

3. MAIN RESULTS

Let $r_{(n)}(t)$ denote the hazard function of X_n, then

$$r_{(2)}(t) = \frac{f_{(2)}(t)}{\bar{F}_{(2)}(t)} = \frac{r(t)R(t)}{1+R(t)} < r(t),$$

for all t, $t > 0$.

In general

$$r_{(n)}(t) < r_{(n-1)}(t) \;, \; n \geq 2 \quad \text{for all } t, \; t > 0$$

where $r_{(1)}(t) = r(t)$.

Let $U_n = X_n - X_{n-1}$, $n = 1, 2, 3, \ldots$ be the time between n-th and (n-1)-th failures. Suppose $G_n(t)$ and $g_n(t)$ be respectively the probability distribution function and probability density function of U_n. Then we can write

$$1 - G_n(t) = \int_0^\infty \frac{\bar{F}(t+u)}{\bar{F}(u)}\, f_{(n-1)}(u)du$$

$$= \int_0^\infty f(t+u)\, \frac{(R(u))^{n-1}}{(n-1)!}\, du \;.$$

Lemma 3.1

If F belongs to c_1, then $E(U_n) \leq (\geq)\; E(U_1)$ according as F is NBU (NWU).

$$E(U_n) = \int_0^\infty \int_0^\infty (\Gamma(n))^{-1}\, (R(u))^{n-1}\, f(u)\, \frac{\bar{F}(z+u)}{\bar{F}(u)}\, dudz$$

$$\leq (\geq) \int_0^\infty \int_0^\infty (\Gamma(n))^{-1} (R(u))^{n-1}\, f(u)\, \bar{F}(z)\, dudz,$$

according as $\bar{F}(z+u) \leq (\geq)\; \bar{F}(u)\, \bar{F}(z)$. Hence $E(U_n) \leq (\geq)\; E(U_1)$.

(a) Uniform Distribution:

Suppose the random variable U_1 has a two parameter rectangular distribution with the following probability density function

$$f(x) = \frac{1}{\beta-\alpha}, \quad -\infty < \alpha < \beta < \infty$$
$$= 0, \quad \text{otherwise.}$$

It can easily be shown that

$$f_n(x) = \frac{1}{(n-1)!}\ \frac{1}{\beta-\alpha}\ (\ln\frac{\beta-\alpha}{\beta-x})^{n-1}, \quad \alpha < x < \beta$$

$$E(X_n) = 2^{-n}\alpha + (1-2^{-n})\beta$$

$$V(X_n) = (3^{-n} - 4^{-n})(\beta-\alpha)^2 .$$

The joint pdf of X_m, X_n $(m < n)$ is

$$f_{m,n}(x_m, x_n) = \frac{1}{(m-2)!}\ \frac{1}{(n-m-1)!}\ \frac{1}{\beta-\alpha}\cdot\frac{1}{\beta-x_m}$$
$$\cdot(\ln\frac{\beta-a}{\beta-x_m})^{m-2}\ (\ln\frac{\beta-x_m}{\beta-x_n})^{n-m-1}$$
$$\alpha < x_m < x_n < \beta$$
$$= 0, \quad \text{otherwise.}$$

$$E(X_n \mid X_m = x_m) = 2^{m-n}\, x_m + (1 - 2^{m-n})\,\beta$$

$$\mathrm{Cov}(X_m, X_n) = 2^{m-n}\,\mathrm{Var}(X_m), \quad n > m.$$

The minimum variance unbiased estimates $\hat{\alpha}$, $\hat{\beta}$ of α and β based on the observed values $x_1, x_2, \ldots, x_n$ of $X_1, X_2, \ldots, X_n$ are

$$\hat{\alpha} = 2x_1 - \hat{\beta}$$

$$\hat{\beta} = \frac{4}{3(3^{n-1}-1)}\,(3^{n-1}x_m - \frac{1}{2}3^{n-2}\cdot x_{m-1} - \ldots - \frac{3}{2}x_1)$$

$$V(\hat{\alpha}) = \frac{1}{9}\,\frac{3^n-1}{3^{n-1}-1}\,(\beta-\alpha)^2$$

$$V(\hat{\beta}) = \frac{2}{9}\,\frac{3^n-1}{3^{n-1}-1}\,(\beta-\alpha)^2$$

$$\mathrm{Cov}(\hat{\alpha}, \hat{\beta}) = -\frac{2}{9}\,\frac{1}{3^{n-1}-1}\,(\beta-\alpha)^2 .$$

The best linear unbiased predictor $\hat{X}_s$ of X_s is

$$\hat{X}_s = 2^{-s}\hat{\alpha} + (1-2^{-s})\hat{\beta} + 2^{n-s}\,(X_n - \frac{\hat{\alpha}}{2^n} - (1-\frac{1}{2^n})\hat{\beta})$$
$$= 2^{n-s}X_n - (2^{n-s}-1)\hat{\beta} .$$

(b) Pareto Distribution:

Suppose the random U_1 has the Pareto distribution with the following cumulative distribution function $F(x)$

$$F(x) = 1 - (\theta/x)^{\nu}, \quad 0 < \theta \le x, \ \nu > 0.$$

The pdf $f_n(x)$ of X_n can be written as

$$f_n(x) = \frac{\nu^n}{(n-1)!}\,\frac{1}{\theta}\,(\frac{\theta}{x})^{\nu+1}\,(\ln(x/\theta))^{n-1},$$
$$0 < \theta \le x < \infty$$
$$= 0, \quad \text{otherwise.}$$

$$E(X_n) = \theta\,(\frac{\nu}{\nu-1})^n, \quad \text{for } \nu > 1$$

$$E(X_n^k) = \theta^k\,(\frac{\nu}{\nu-k})^n, \quad \nu > k$$

$$X_n \overset{d}{=} \prod_{i=1}^{n} U_i, \quad n \ge 1$$

where $\overset{d}{=}$ denotes the equality in distribution and $U_1, U_2, \ldots, U_n$ are independent and identically distributed as Pareto distribution with the follows pdf $f(x)$, where

$$f(x) = \nu\, x^{-(\nu+1)}, \quad x \ge 0$$
$$= 0, \quad \text{otherwise.}$$

The product moments of X_m and X_n $(m < n)$ can be obtained as follows

$$X_m^r X_n^s \overset{d}{=} \theta^{r+s}\,(\prod_{i=1}^{m} U_i^{r+s})\,(\prod_{i=m+1}^{n} U_i^{s})$$

and thus

$$E(X_m^r X_n^s) = \theta^{r+s}\,(\frac{\nu}{\nu-r-s})^m\,(\frac{\nu}{\nu-1})^{n-m}$$

Hence

$$E(X_m X_n) = \theta^2(\frac{\nu}{\nu-2})^m\,(\frac{\nu}{\nu-1})^{n-m}$$

$$\mathrm{Cov}(X_m, X_n) = \theta^2\,(\frac{\nu}{\nu-1})^{n-m}\,\mathrm{Var}(X_m)$$

Table 1: Variances and Covariances of X_m, X_n

m	n	α = 2.5	3	3.5	4.0	4.5	5.0
1	1	2.2222	.7500	.3733	.2222	.1469	.1042
1	2	3.7037	1.1250	.5227	.2963	.1889	.1303
2	2	17.2840	3.9375	1.6028	.8395	.5074	.3364
1	3	6.1728	1.6875	.7317	.3951	.2429	.1628
2	3	28.8067	5.9063	2.2440	1.1193	.6524	.4205
3	3	103.5665	15.6094	5.1742	2.3813	1.3148	.8149
1	4	10.2881	2.5313	1.0244	.5267	.3123	.2035
2	4	48.0110	8.8594	3.1416	1.4925	.8387	.5256
3	4	172.6109	23.4141	7.2438	3.1751	1.6905	1.0187
4	4	565.5626	55.3711	14.8841	6.0113	3.0304	1.7556

Table 1 gives the variances and covariances of X_m, X_n for $\nu = 2.5, 3, 3.5, 4, 4.5, 5.0$ and $1 \leq m, n \leq 4$ with $\theta = 1$.

(c) Exponential Distribution:

Suppose the random variable U_1 has a two parameter exponential distribution with the following pdf f(x)

$$f(x) = \sigma^{-1} \exp(-\sigma^{-1}(x-\mu)), \quad \text{for } x > \mu,\ \sigma > 0,$$

$$= 0, \quad \text{otherwise.}$$

The pdf $f_n(x)$ of X_n can be written as follows:

$$f_n(x) = \frac{(x-\mu)^{n-1}}{n} \cdot \frac{1}{\sigma^n} e^{-\sigma^{-1}(x-\mu)}, \quad x > \mu .$$

$$E(X_n) = \mu + n\sigma$$

$$V(X_n) = n\sigma$$

$$Cov(X_n, X_m) = m\sigma^2, \ m < n .$$

The minimum variance unbiased estimates $\hat{\mu}$, $\hat{\sigma}$ of μ and σ are

$$\hat{\mu} = (mx_1 - x_m)/(m-1)$$

$$\hat{\sigma} = (x_m - x_1)/(m-1)$$

$$Var(\hat{\mu}) = m\sigma^2/(m-1)$$

$$Var(\hat{\sigma}) = \sigma^2/(m-1)$$

$$Cov(\hat{\mu}, \hat{\sigma}) = -\sigma^2/(m-1) .$$

The best linear unbiased lenear predictor $\hat{X}_s$ of X_s based on the observed failure times x_1, x_2, ..., x_n is

$$\hat{X}_s = ((s-1)x_m - (s-m)x_1)/(m-1)$$

$$E(\hat{X}_s) = \mu + s\sigma - E(X_{L(s)})$$

$$V(\hat{X}_s) = \sigma^2((m+s^2) - 2s)/(m-1)$$

REFERENCES

1. Ahsanullah, M. and Kabir, A.B.M.L. (1973). A characterization of the Pareto distribution. Can. J. Statist. 1, 109-112.

2. Arnold, B. C. (1983). Pareto distribution. International Publishing Company, Burtonsville, Maryland.

3. Carlton, A. G. (1946). Estimating the parameters of a rectangular distribution. Annals of Mathematical Statistics, 17, 355-358.

4. Goldberger, A. S. (1962). Best linear unbiased prediction in the generalised linear regression model. J. Amer. Statist. Ass. 57, 369-375.

THE SIMULATION OF LIFE TESTS WITH RANDOM CENSORING

Joseph C. Hudson, GMI Engineering & Management Institute

Abstract

This paper considers the simulation of life tests in which n items are placed on test and remain until removed by either failure or random censoring. The censoring mechanism is taken to be independent of the failure mechanism. Simulation is done under the constraint that the number of items censored is a Binomial random variable, allowing simulations to be run specifying the expected percentage of censored items.

Details of the implementation are discussed and a validation study is presented. The simulation is implemented in Pascal.

Introduction

The development of techniques for reliability data analysis requires data from known distributions for empirical validation and comparison studies. Such a need motivated the work reported in this paper. Randomly censored failure data was needed from a spectrum of short and long tailed distributions. The algorithm presented here simulates data from tests in which n items are placed on test. Each item remains on test until either failure or removal from test by a random censoring mechanism independent from the failure mechanism.

Simulations are carried out using Weibull, uniform, truncated normal and truncated Cauchy failure distributions. The censoring distribution is taken to be exponential. With user-specified failure distribution and probability of censoring p_c, the mean of the censoring distribution is determined to enforce the constraint that $P(T_{ci} < T_{fi}) = p_c$, where T_{ci} and T_{fi} are the censoring and failure times of the i^{th} item, respectively. In performing the simulation, a failure time and a censoring time are independently generated for each item, with the smaller of these times taken as the time of removal from test. Time of and reason for removal from test are reported for each item.

Use of the simulation procedure involves the following steps:

1. Choose a failure distribution from the truncated Cauchy, truncated normal, uniform or Weibull families.
2. Choose a probability of censoring and find the mean of the censoring distribution.
3. Choose the sample size n. Generate the random sample using the following procedure for each item:
   a. Randomly generate a failure time.
   b. Randomly generate a censoring time.
   c. The smaller of the two times determines the stopping event, failure or censoring. Record the type of event and the time of occurrance.

Each of these steps will be discussed below.

Choice of Failure Distribution

The distributions available were chosen to offer both long and short tailed alternatives to the Weibull. The Cauchy and normal distributions are truncated at 0 to avoid negative failure times. Since step 2 is implemented with general procedures, the list of available failure distributions can be expanded.

Finding the Mean of the Censoring Distribution

The censoring time for the i^{th} item on test, T_{ci}, follows the exponential distribution with density

$$f_c(t) = \frac{1}{\mu} e^{-t/\mu}, \quad t \geq 0.$$

For brevity, the i subscript is supressed. For given μ and failure CDF $F_f(t)$, the probability that the i^{th} event is a censoring is

$$P(\mu) = P(Tc < T_f) = \int_0^\infty P(T_f > t) f_c(t) dt$$

$$= \frac{1}{\mu} \int_0^\infty (1 - F_f(t)) e^{-t/\mu} dt. \qquad (1)$$

A representative graph of $P(\mu)$ is shown in figure 1. $P(\mu)$ has a number of useful properties:

$$\lim_{\mu \to 0^+} P(\mu) = 1$$

$$\lim_{\mu \to \infty} P(\mu) = 0 \qquad (2)$$

$P(\mu)$ is monotonically decreasing in μ.

Proof is straightforward. These properties guarantee a unique solution μ_c to the equation

$$P(\mu) = p_c. \qquad (3)$$

μ_c is found using the secant method (Hornbeck, [1975]) modified as shown in figure 2. The modification involves the behavior of the points (μ_1, p_1) and (μ_2, p_2) used to define the secant line. The relationship $p_1 > p_c$ is maintained to keep the point (μ_1, p_1) to the left of the goal (μ_c, p_c). This insures that the sequence of μ_1 values monotonically approaches μ_c, at the expense of computation time. The procedure

terminates when the relative difference between μ_1 and μ_2 falls below a specified error tolerance.

Each iteration of the secant method requires evaluation of (1) to find p_2. We briefly discuss the procedure for doing this for each of the failure distributions.

If T_f has a uniform distribution on [a,b], $a \geq 0$,

$$P(\mu) = 1 + \frac{\mu}{b-a}\left[e^{-b/\mu} - e^{-a/\mu}\right]. \quad (4)$$

Solution of (3) for (μ_c, p_c) proceeds without difficulty.

If T_f has a truncated normal distribution with (pre truncation) mean M and standard deviation σ,

$$P(\mu) = 1 - \left[\mu/\left(\sigma(1-P_N)\sqrt{2\pi}\right)\right]\int_0^{\infty} e^{-t} e^{-(\mu t - M)^2/2\sigma^2} dt. \quad (5)$$

where P_N is the probability that a normal random variable with mean M and standard deviation σ is negative. (5) may be reduced to

$$P(\mu) = 1 - P(Z > A) e^{-M/\mu + \sigma^2/2\mu^2}/(1-P_N) \quad (6)$$

where $A = \sigma/\mu - M/\sigma$ and Z is the standard normal variate. (6) may be readily evaluated using an adaption of Maclaurin series and continued fraction expansions of the error function (Nonweiler [1984]). Hudson [1986] gives details.

If T_f has a three parameter Weibull distribution with minimum life δ, characteristic life θ and shape parameter β, then

$$P(\mu) = 1 - e^{-\delta/\mu}\left[1 - \int_0^{\infty} e^{-t} e^{-(\mu t/\theta)^{\beta}} dt\right]. \quad (7)$$

The integral in (7) may be replaced with an integral with finite limits using the relationship

$$\int_0^{\infty} e^{-t} e^{-(\mu t/\theta)^{\beta}} dt \approx \int_0^{-\ln(2E)} e^{-t} e^{-(\mu t/\theta)^{\beta}} dt + E. \quad (8)$$

The error of approximation using (8) is less than E, so (3) may be solved for μ_c to the desired error tolerance by assigning a portion of the error tolerance to E. The numerical evaluation of the integral in (8) is carried out using adaptive Simpson's quadrature with Richardson's improvement (Marion [1987]). The variant used is shown in figure 3.

If T_f has a truncated Cauchy distribution with pretruncation median a and scale parameter b, then

$$P(\mu) = 1 - \frac{\text{Atan}(a/b)}{\pi(1-P_N)} - \frac{1}{\pi(1-P_N)}\int_0^{\infty} e^{-t}\,\text{Atan}\left(\frac{\mu t - a}{b}\right) dt. \quad (9)$$

where Atan is the inverse tangent function and P_N is the probability that the pretruncation failure random variable is negative. This case is processed as the Weibull, with the integral in (9) approximated by

$$\int_0^{x} e^{-t}\,\text{Atan}\left(\frac{\mu t - a}{b}\right) dt + .5e^{-x}\left(\frac{\pi}{2} + \text{Atan}\left(\frac{\mu x - a}{b}\right)\right) \quad (10)$$

where $x \geq a/\mu$ satisfies the inequality

$$E \leq .5e^{-x}\left(\frac{\pi}{2} - \text{Atan}\left(\frac{\mu x - a}{b}\right)\right) \quad (11)$$

with E the desired error tolerance in the integral approximation. x is found using the secant method.

Generating the Random Samples

The sampling procedure requires n random observations each from the censoring and failure distributions. To generate these, the output of a uniform [0,1) random number generator is shuffled using algorithm B of Knuth [1981, pg32] with an auxiliary table of length 117. The resulting uniform [0,1) random number is converted as needed using standard transformations for the uniform, Weibull and Exponential cases (Hastings and Peacock, [1975]). Normal deviates are generated using a ratio method, algorithm R of Knuth [1981, pg 125]. Cauchy variates are generated using the ratio of two independent standard normal deviates (Hastings and Peacock, [1975, pg 42]).

Validation Study

The algorithm is implemented in Pascal. To verify the implementation, 100 samples of 100 items each were generated for each of the 24 failure distributions shown in table 1. The estimated value of P_c generated by the samples and the value of μ_c found by the algorithm and used in the sampling procedure are also shown.

If the sampling procedure performs as designed, each of the 24 sets of 100 samples is a sample of size 100 from a binomial distribution with n = 100 and p the probability of censoring, either .1 or .9. Chi square goodness of fit tests were carried out to test this hypothesis against its negation. 11 cells were used for each test, giving 10 degrees of freedom. The resulting χ^2 values and their P values are shown in table 1. Figures 4 and 5 show the best and worst cases from among the 24.

The 24 observations of χ^2 should constitute a random sample from the chi square distribution with 10 degrees of freedom if the algorithm is properly implemented. An additional study of the ordered χ^2 values did not reveal any grouping or unusual patterns among these 24 values.

References

Hastings, N. A. J. and J. B. Peacock. 1975. Statistical Distributions. A Handbook for Students and Practitioners. Butterworth & Co. London.

Hornbeck, R. W. 1975. Numerical Methods. Quantum Publishers, Inc. New York.

Hudson, J. C. 1986. Computing the Standard Normal CDF. submitted for publication.

Knuth, D. E. 1981. The Art of Computer Programming. Volume 2. Seminumerical Algorithms. Addison Wesley, Reading, MA.

Nonweiler, T. R. F. 1984. Computational Mathematics. Halstead Press, John Wiley & Sons. New York.

Marion, M. J. 1987 Numerical Analysis. Macmillan, New York.

Table 1. Summary of the validation study.

Dist	Ref No.			P_c	Sample Est of P_c	μ_c	χ^2 10df	P Value
Cauchy		Median	Shape					
	1	10,000	2	.1	.1003	94,928	8.51	.579
	2	10,000	2	.9	.8997	4,342	16.87	.077
	3	10,000	10,000	.1	.1000	229,378	5.54	.852
	4	10,000	10,000	.9	.8996	3,482	8.31	.599
Normal		Mean	St Dev					
	5	10,000	2	.1	.1014	94,912	12.76	.237
	6	10,000	2	.9	.8985	4,343	3.13	.978
	7	10,000	10,000	.1	.1046	119,746	6.99	.726
	8	10,000	10,000	.9	.9022	2,810	6.66	.757
Uniform		MinLife	MaxLife					
	9	0	10,000	.1	.1009	46,608	9.19	.514
	10	0	10,000	.9	.9089	1,000	17.80	.058
	11	9,900	10,000	.1	.1016	94,438	7.76	.652
	12	9,900	10,000	.9	.9011	4,321	6.54	.768
Weibull		MinLife	Slope	CharLife = 10,000				
	13	0	0.5	.1	.0959	149,074	8.50	.580
	14	0	0.5	.9	.9016	146	7.36	.691
	15	0	1.5	.1	.0981	83,604	2.19	.995
	16	0	1.5	.9	.8946	2,005	8.45	.585
	17	0	4.0	.1	.0982	85,670	4.72	.909
	18	0	4.0	.9	.8985	3,536	16.07	.098
	19	9,900	0.5	.1	.1005	256,214	8.89	.543
	20	9,900	0.5	.9	.9029	6,358	15.21	.125
	21	9,900	1.5	.1	.1003	178,659	3.19	.977
	22	9,900	1.5	.9	.9004	7,370	6.83	.741
	23	9,900	4.0	.1	.1005	179,821	8.41	.780
	24	9,900	4.0	.9	.8975	8,061	8.51	.579

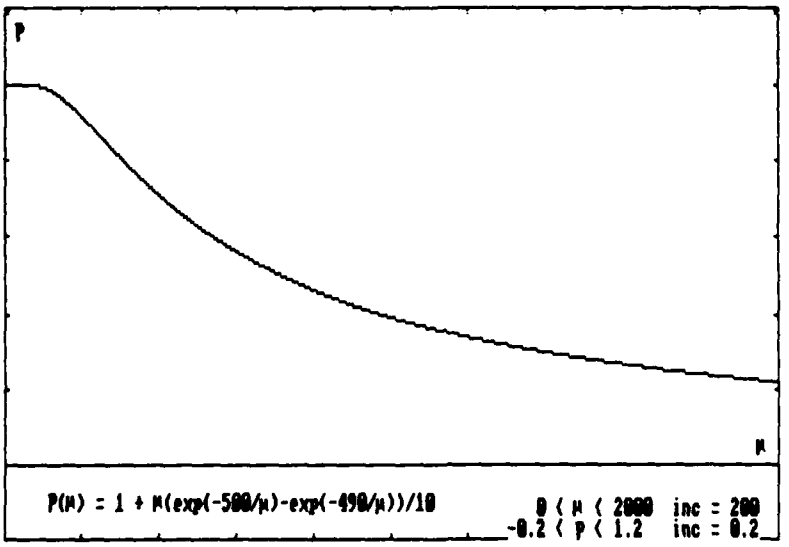


Figure 1. P(μ) for the uniform [490,500] failure distribution.

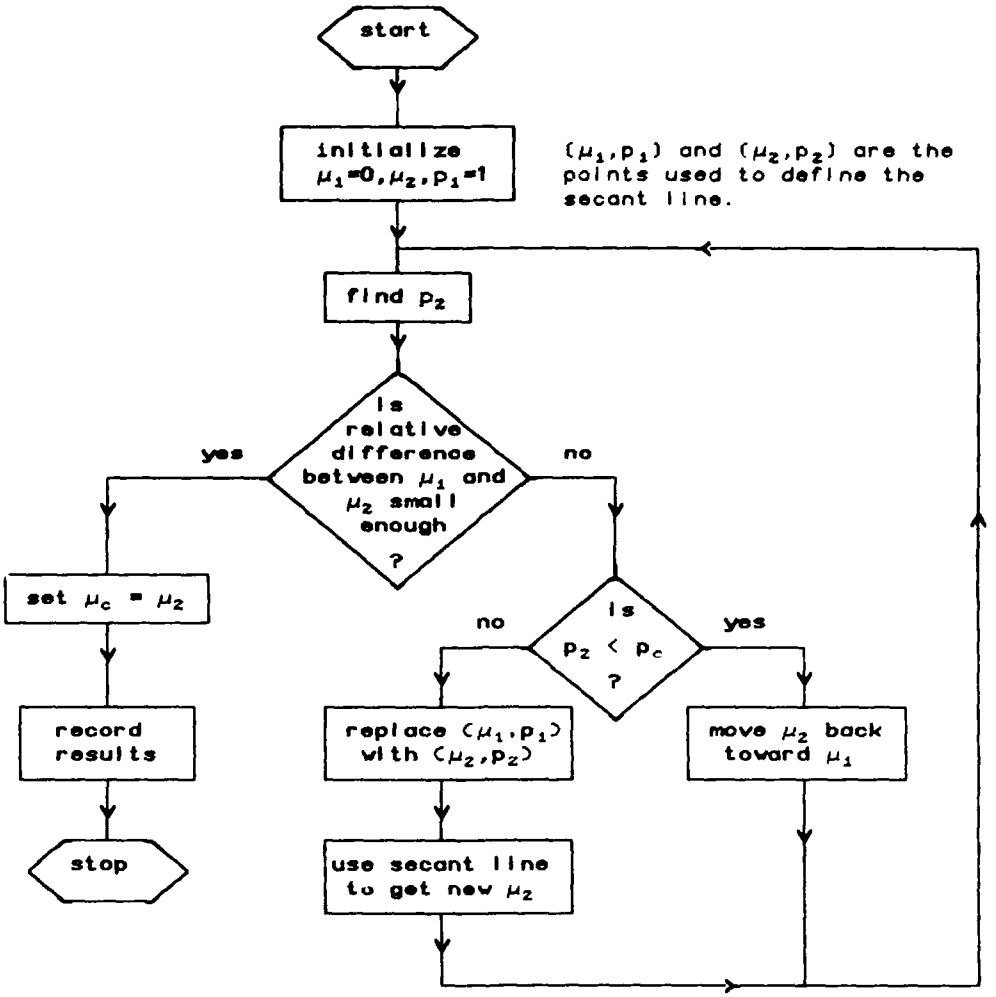


Figure 2. Modified secant method used to find μ_c.

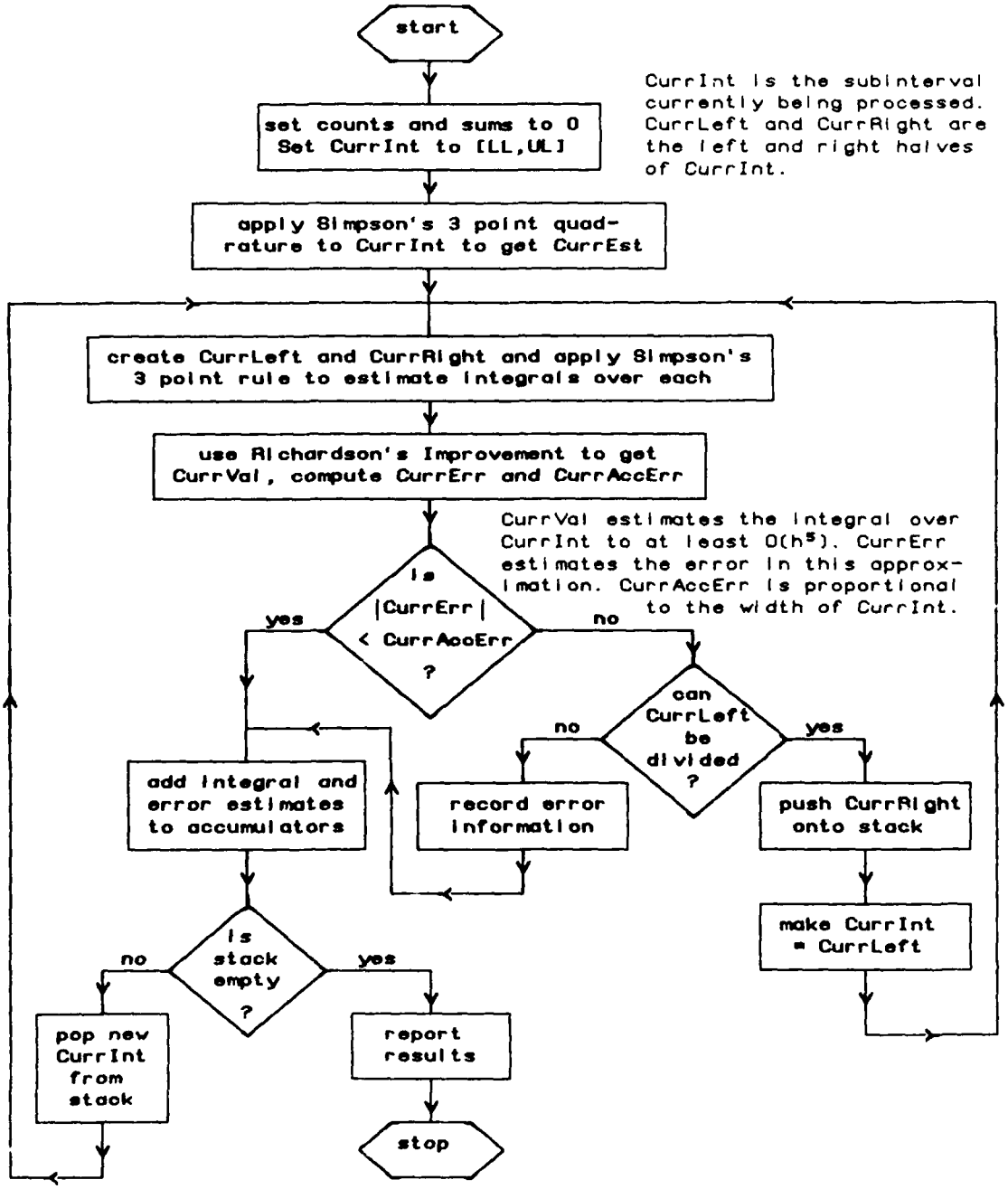


Figure 3. Adaptive Simpson's Quadrature applied to $\int_{LL}^{UL} f(t)dt$.

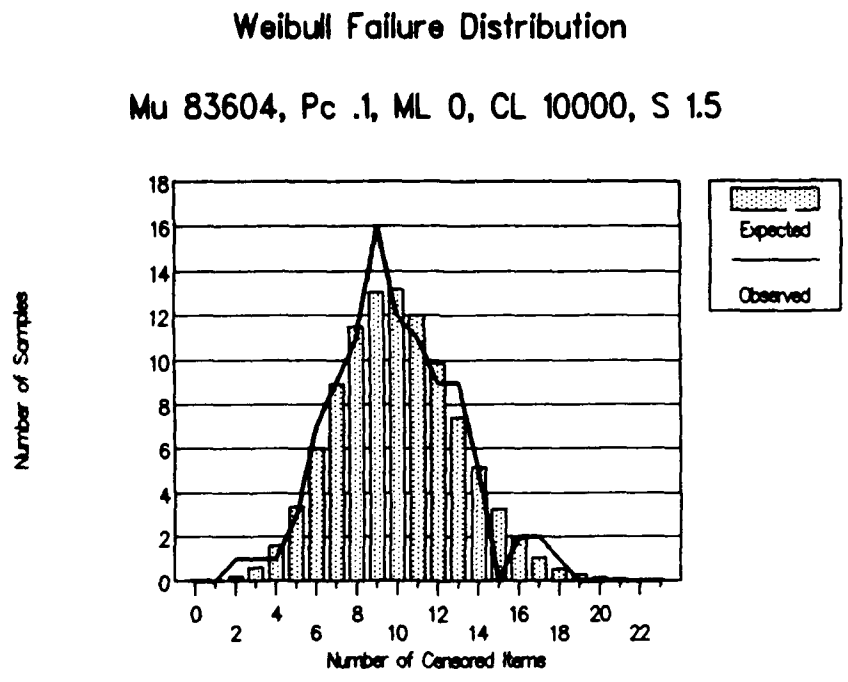


Figure 4. The best fit from the validation study.

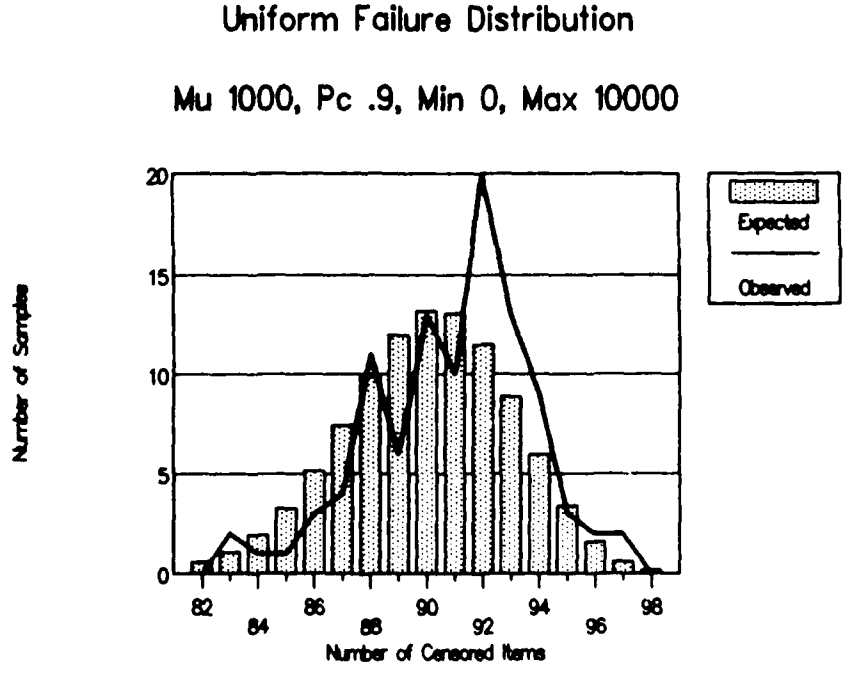


Figure 5. The worst fit from the validation study.

An Identifiable Model for Informative Censoring

William A. Link, U.S. Fish and Wildlife Service, Patuxent Wildlife Research Center

1. INTRODUCTION

The usual model for censored survival analysis of a "lifetime" T is that observations are of the form (X,δ) where $X \leq T$ and δ takes the values 0 and 1, depending on whether $X < T$ or $X = T$, respectively. A great deal of attention has been given to the "independent censoring model" in which it is assumed that $X = \min(T,C)$, with C (referred to as the "censoring variable") assumed to be statistically independent of the lifetime under consideration. This model is appealing because of its qualitative simplicity and mathematical tractability.

Under the independent censoring model, the Kaplan-Meier estimator (KME) (Kaplan and Meier, 1958) is the appropriate estimator of the survival function $S(t) = P(T > t)$. Földes and Rejtö (1981) established strong uniform consistency of the KME and Gill (1983) provided weak convergence results on the entire positive half-line.

Analyses based on the usual model may yield unreliable results if the independent censoring assumption is inappropriate. As a hypothetical example, consider a radio telemetry study of the life expectancy of a mallard. Censored observations would occur upon failure of the radio transmitter. If failure of the transmitter were related solely to the reliability of the unit, the event of censoring could safely be assumed to carry no information about the status of the bird. However if failure of the unit were due to predation, censoring would be equivalent to death of the bird. Between these two extremes is a wide range of possible models in which the event of censoring carries an unfavorable prognosis for future survival. In such a case the KME will tend to overestimate the true survival probabilities.

Unfortunately, if the only observations available are the pairs (X,δ), the independence assumption is completely untestable. It has been shown by Cox (1959) and Tsiatis (1975) that "there always exist independent censoring models consistent with any probability distribution for the observable pair (X,δ)" (Lagakos, 1979). Consequently, if the independence assumption is deemed inappropriate, analysis must rely on equally untestable model assumptions and (perhaps) the observation of covariates.

The reader is referred to the work of Williams and Lagakos (1977) and Lagakos and Williams (1978) as well as that of Robertson and Uppuluri (1984) for examples of such methods. In the former, it is assumed that the hazard function of the X's is related to that of the T's by

$$\lambda_X(t) = \frac{c(t) + \theta}{c(t)} \lambda_T(t) ,$$

where

$$c(t) = \frac{P(\delta = 1 | X = t)}{P(\delta = 0 | X = t)}$$

They suggest approximating $c(t)$ by a step function taking k distinct values $c_1,\ldots,c_k$. Their procedure for estimating $S(\cdot)$ under the cone-class model assumption involves an arbitrary choice of the number and length of these intervals and, in addition, specifying that $S(\cdot)$ is a member of some parametric family of distributions. Estimation of the parameters corresponding to $S(\cdot)$, of $c_1,\ldots,c_k$, and of θ, is then accomplished by maximum likelihood. Explicit solution of the maximum likelihood equations is not possible; their method "requires a great deal of computer time."

Another procedure for obtaining alternative estimators to the KME has been proposed by Robertson and Uppuluri (1984). Their procedure is based on a modification of the well-known "redistribute to the right" algorithm (Efron, 1967). There do not appear to be easily constructible models justifying most redistribution schemes.

There would appear to be a need for some simple models, and corresponding survival function estimators, applicable when censoring carries an unfavorable prognosis for future survival. In this paper we propose a model in which censoring can occur only in a "high-risk" subpopulation. This model suggests a modification of Efron's self-consistency algorithm which leads to our Modified Kaplan-Meier estimator (MKME).

2. THE MODEL

Suppose that a lifetime "T" with survival function S(t) is the object of investigation, that for each lifetime there is a binary covariate "γ", with

$$Q(t) \equiv P\left(T > t \mid \gamma = 1\right) = \left(S(t)\right)^m$$

for some $m \geq 1$. It follows that

$$R(t) \equiv P\left(T > t \mid \gamma = 0\right) = \frac{S(t) - p\,Q(t)}{1 - p} ,$$

where $p = P(\gamma = 1)$. Since $-dR(t)/dt$ must be non-negative, the values of m and p are restricted by $mp \leq 1$.

In this model, lifetimes with covariate $\gamma = 1$ have a hazard function $\lambda_Q(\cdot) = m\lambda(\cdot)$, where $\lambda(\cdot)$ is the population hazard function. Thus the covariate γ divides the population into "high-risk" and "low-risk" subpopulations.

Let $(T_1,\gamma_1),(T_2,\gamma_2),\ldots,(T_n,\gamma_n)$ be a sample of these lifetimes and their corresponding covariates. Furthermore, let $C_1,C_2,\ldots,C_n$ be a sample of "potential censoring times," independent of the corresponding T's. By this we mean that observations are of the form

$$X_i = \left(1 - \gamma_i\right) T_i + \gamma_i \min\left\{T_i , C_i\right\}$$

and (2.1)

$$\delta_i = \left(1 - \gamma_i\right) + \gamma_i I\left(T_i \leq C_i\right) ,$$

where $I(\cdot)$ is the indicator function. Thus censoring is only a possibility in the high risk subpopulation.

Use of the KME under this model leads to overestimates of S(t) (see §5). In the sequel we shall consider an alternative estimator based on a modification of Efron's Self Consistency Algorithm which is appropriate under this model.

3. SURVIVAL FUNCTION ESTIMATOR

Letting $0=x_0<x_1<x_2<\cdots<x_n$ represent the ordered times of observation and $\delta_{(1)},\delta_{(2)},\ldots,\delta_{(n)}$ represent the corresponding values of δ, the KME is the unique limit (as $K \to \infty$) of the sequence of functions obtained by

$$\tilde{S}^{(K+1)}(t) = \frac{1}{n}\left\{\sum_{i=1}^{n} I\left(x_i > t\right) + \sum_{i=1}^{n}\left(1-\delta_{(i)}\right)\frac{\tilde{S}^{(K)}(t)}{\tilde{S}^{(K)}(x_i)}\right\}.$$

Thus the KME satisfies

$$\tilde{S}(t) = \frac{1}{n}\left\{\sum_{i=1}^{n} I\left(x_i > t\right) + \sum_{i=1}^{n}\left(1-\delta_{(i)}\right)\frac{\tilde{S}(t)}{\tilde{S}(x_i)}\right\},$$

which is to say that the estimated probability of survival beyond time "t" is the percentage of observations (censored or uncensored) beyond time "t" plus the estimated percentage that would have survived beyond time "t" but were censored before "t". The KME is said to be "self-consistent" because, in the independent censoring model, for $x_i < t$,

$$P\left(T > t \mid X = x_i,\ \delta = 0\right) = \frac{S(t)}{S(x_i)} .$$

Under the model discussed in §2,

$$P\left(T > t \mid X = x_i,\ \delta = 0\right) = \left\{\frac{S(t)}{S(x_i)}\right\}^{m} ,$$

suggesting that the self-consistency algorithm be replaced by

$$\tilde{S}^{(K+1)}(t) = \frac{1}{n}\left\{\sum_{i=1}^{n} I\left(x_i > t\right) + \sum_{i=1}^{n}\left(1-\delta_{(i)}\right)\left(\frac{\tilde{S}^{(K)}(t)}{\tilde{S}^{(K)}(x_i)}\right)^{m}\right\}.$$

The following facts regarding this sequence of survival function estimators are proved by Link (1986):

1) For each fixed m, the sequence thus defined converges. Furthermore, if the original estimator $\tilde{S}^{(0)}$ places no weight on censored observations or beyond the range of observations, the limit is unique. This will be called the MKME, and denoted by $\tilde{S}_m$.

2) For $m_1 \le m_2$, $\tilde{S}_{m_1}(t) \ge \tilde{S}_{m_2}(t)$. Thus the KME (obtained using m=1) bounds these estimators from above, while the empirical survival function of the X's (obtained using large values of m) bounds them from below.

4. ESTIMATION OF MODEL PARAMETERS

In this section we shall show that if the covariate γ is observable, an estimate of the parameter m is available.

From the definition of X (2.1) we find that

$$P\left(S(X) \le t \mid \gamma = 0\right) = \frac{t - p\,t^{m}}{1-p},$$

$t \in (0,1)$.

Letting $c = E\left(S(X)\mid\gamma=0\right)$, we have

$$c = \int_0^1 P\left(S(X) > t \mid \gamma = 0\right) dt$$

$$= 1 - (1-p)^{-1}\left(\frac{1}{2} - \frac{p}{m+1}\right) . \qquad (4.1)$$

Given a value of m, the parameter c can be estimated by

$$\hat{c}(m) = \frac{\sum_{j=1}^{n} \gamma_{(j)}\ \tilde{S}_m(x_j)}{\sum_{j=1}^{n} \gamma_{(j)}} ,$$

where $\gamma_{(j)}$ is the value of γ corresponding to x_j, and $\tilde{S}_m(\cdot)$ is the MKME obtained using the given value of m. The parameter p is estimated by $\hat{p} = \Sigma\gamma/n$, so that given a value of c, and using (4.1), an estimate of m can be obtained as

$$\hat{m}(c) = \frac{2\hat{p}}{1 - 2c(1-\hat{p})} - 1 .$$

It is easily verified that there exists uniquely a pair (m_o,c_o) satisfying simultaneously $\hat{c}(m_o) = c_o$ and $\hat{m}(c_o) = m_o$. These can be found by a variety of numerical methods, such as repeated substitution. Since the model restrictions of §2 require $mp \le 1$, we suggest the use of

$$\tilde{m} = \min\left\{ m_o ,\ (\hat{p})^{-1} \right\} ,$$

as the estimator of the model parameter.

5. SIMULATION RESULTS

Rather than consider a specific survival function $S(\cdot)$, we generated pairs (U_T,γ) where γ is a Bernoulli variable with parameter p, and where U_T satisfies

$$P(1 - U_T \le x \mid \gamma = 1) = x^{m} ,$$

and

$$P(1 - U_T \le x \mid \gamma = 0) = \frac{x - (1-p)\,x^{m}}{p} ,$$

$0 \le x \le 1$.

The U_T's can be thought of as the quantiles of a random sample from an arbitrary continuous survival function $S(\cdot)$. Observations of the form (U_X,δ) were then obtained, where

$$U_X = \left(1-\gamma\right)U_T + \gamma \min\left\{U_T ,\ U_C\right\} ,$$

and

$$\delta = 1 - \gamma\, I\left(U_T > U_C\right) ,$$

and the potential censoring variable U_C, generated independently of U_T and γ, satisfies $P(U_C > x) = x^{\alpha}$, $0 < x < 1$, where $\alpha > 0$ is a specified constant.

In order to investigate the large sample behavior of the MKME, a sample of size 1000 was generated using $p = .5$, $m = 2$, and $\alpha = .625$, yielding an expected censoring rate $\approx .2344$. The results are summarized in Table 1. The quantiles of the KME obtained for these data are also included. It is seen that under this model the KME seriously overestimates true

survival probabilities.

Table 1. *Estimates of quantiles of* $F(T)$, m=2, p=.4, n= 1000

t	*kme*	*mkme*	t	*kme*	*mkme*
.05	.0444	.0470	.55	.4758	.5530
.10	.0922	.1001	.60	.5211	.6003
.15	.1362	.1510	.65	.5581	.6377
.20	.1738	.1961	.70	.6155	.6930
.25	.2226	.2572	.75	.6762	.7485
.30	.2731	.3177	.80	.7299	.7950
.35	.3127	.3656	.85	.7779	.8348
.40	.3574	.4189	.90	.8612	.9002
.45	.3834	.4494	.95	.9284	.9498
.50	.4266	.4989			

In addition, a limited Monte Carlo study was carried out to investigate the sampling distribution of $\hat{m}$. The estimates given in Table 2 were obtained by generating 100 samples of size 100 in the manner described above. It appears that the sampling distribution of $\hat{m}$ is skewed to the right and that $\hat{m}$ tends to slightly overestimate m.

Table 2. *Estimates of Mean & Standard Deviation of* $\hat{m}$. (n=100)

m	p	*Censoring Rate*	*Mean*	*St. Dev.*
1.25	0.50	0.284	1.318	0.182
1.50	0.50	0.247	1.564	0.204
1.75	0.50	0.248	1.819	0.235
2.00	0.75	0.117	2.223	0.498
2.00	0.50	0.234	2.106	0.306
3.00	0.70	0.116	3.414	0.730

6. DISCUSSION

The model considered in this article offers an alternative to the usual independent censoring model. Censoring is a possibility only in a subpopulation whose hazard function is m times that of the population at large. The parameter m needs only to be non-negative; values of $m < 1$ describe models in which the event of censoring carries a favorable prognosis for further survival.

If the covariable γ is not observed, a (weak) upper bound on the range of possible values of m can be obtained by noting that

$$\frac{1}{P(\delta = 0)} \geq \frac{1}{P(\gamma = 1)} \geq m ,$$

and estimating $P(\delta = 0)$ in the obvious way.

The author wishes to thank Christine Bunck, Nancy Coon, Paul Geissler, Thomas Mathew, and Kenneth Pollock for their careful review and valuable editorial suggestions.

REFERENCES

Cox, D. R. (1959), "The Analysis of Exponentially Distributed Life-times with Two Types of Failure," *Journal of the Royal Statistical Society, Series B* 59, 411-421.

Efron, B. (1967), "The Two Sample Problem with Censored Data," *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol.* 4, 831-853.

Földes, A. and Rejtö, L. (1981), "Strong Uniform Consistency of the Product-Limit Estimator under Variable Censoring, *Zeit. Wahr.*, 58, 95-108.

Gill, R. (1983), "Large Sample Behavior of the Product-Limit Estimator on the Whole Line," *Annals of Statistics*, 11, 49-58.

Kaplan, E. L. and Meier, P. (1958), "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, 53, 457-481.

Lagakos, S. W. (1979), "General Right Censoring and its Impact on the Analysis of Survival Data," *Biometrics*, 35, 139-156.

Lagakos S. W. and WIlliams J. S. (1978), "Models for Censored Survival Analysis: A Cone Class of Variable-sum Models," *Biometrika*, 65, 181-189.

Link, W. A. (1986), "Contributions to Reliability Theory and Survival Analysis," unpublished Ph.D. thesis, University of Massachusetts at Amherst, Dept. of Mathematics and Statistics.

Robertson, J. B. and Uppuluri, V. R. R. (1984), "A Generalized Kaplan-Meier Estimator," *Annal of Statistics*, 12, 1, 366-371.

Tsiatis, A. (1975), "A Nonidentifiability Aspect of the Problem of Competing Risks," *Proceedings of the National Academy of Science*, 72, 20-22.

Williams, J. S. and Lagakos S. W. (1977), "Models for Censored Survival Analysis: Constant-sum and Variable-sum Models," *Biometrika*, 64, 215-224.

XVII. APPLICATIONS

Nonparametric Regression and Spatial Data: Some Experiences Collaborating with Biologists
Douglas Nychka, North Carolina State University

Space Balls! Or Estimating the Diameter Distribution of Monosize Polystyrene Microspheres
Susannah B. Schiller, National Bureau of Standards

Maximum Queue Size and Hashing with Lazy Deletion
Claire M. Mathieu, Princeton University; Jeffrey Scott Vitter, Brown University

Classifying Linear Mixtures, with an Application to High Resolution Gas Chromatography
William S. Rayens, University of Kentucky

Bias of Animal Population Trend Estimates
Paul H. Geissler, William A. Link, U.S. Fish and Wildlife Service

The Elimination of Quantization Bias Using Dither
Douglas M. Dreher, Martin J. Garbo, Hughes Aircraft Company

An Alternate Methodology for Subject Database Planning
Henry D. Crockett, Mark E. Eakin, Craig W. Slinkman, University of Texas at Arlington

Sensitivity Analysis of the Herfindahl–Hirschman Index
James R. Knaub, Jr., U.S. Department of Energy

Encoding and Processing of Chinese Language—A Statistical Structural Approach
Chaiho C. Wang, U.S. Department of Justice and George Washington University

NONPARAMETRIC REGRESSION AND SPATIAL DATA: SOME EXPERIENCES COLLABORATING WITH BIOLOGISTS.

Douglas Nychka, North Carolina State University

1 Introduction

The widespread use by scientists of personal computers for data collection and analysis gives the statistician more oppurtunity to become closely involved in a research project. This paper will describe two projects in which I have collaborated with biologists. The main point is a simple one. Computer resources can significantly improve a collaborative relationship between a statistician and an experimenter. This can happen in at least two ways: 1) special purpose statistical software can be developed to guide experimenters in analyzing their data 2) the statistician can be involved in the data collection by helping to develop the software used to collect and store the data.

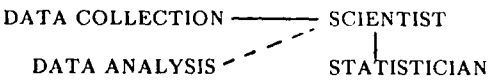
This discussion will be organized by considering a simple model (in the social science sense) that outlines the interaction between the statisitician and the scientist. The next section briefly discusses the components of this model and Sections 3 and 4 give specific examples from two research projects. The first project concerns the estimation of fitness surfaces for a song sparrow population based on the sparrows ability to survive over the winter. In this case the fitness surface is the probability of a sparrow's survival as a function of several body measurements. One success in this project was making some specific nonparametric regression software available to the biologist so he could carry out most of the analysis of his data on his own. The second project studies the spatial distribution of air plants (epiphytes) in the canopy of Costa Rican rain forests. This analysis depends on constructing a three-dimensional "map" of the canopy trees and of the locations of epiphytes using numerous sightings from a transit. I have participted in the data collection by developing software for a PC that estimates the xyz coordinates of points in the canopy from the raw angular measurements. It is important to be able to generate these tree maps right after a day of field work because they serve as a check on the measurements and will direct further subsampling of the trees' branches. Another aspect of this project is to involve this botanist directly in the spatial analysis of the canopy data. One way of accomplishing this goal is to design a small set of macros and compiled functions that make it possible to carry out of the anlaysis witnin the S statistical package.

Although these two projects are used as examples of collaborative work, they both contain novel statistical applications that are of interest in their own right. A reader who is not particularly interested in collaborative aspects is still encouraged to look at Sections 3 and 4 for their statistical content.

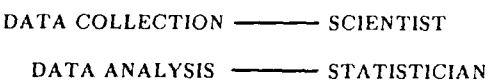
2 Role of the biologist and the statistician

Several different roles of the statistician and the biologist are shown schematically in Figure 1. The relationship shown in the first diagram is a model for short term statistical consulting. The scientist both collects and analyzes his/her data following the advice of a statistician. Here the statistician plays a passive role and becomes involved in the research project only through the scientist. The next diagram is an improvement and indicates the roles often filled by these two people in a collaborative effort. The scientist concentrates on data collection while the statistician is mainly involved in the statistical analysis of the data. There are two potential disadvantages with this separation. The statistician may not gain a true appreciation for the data that have been colected while the analysis performed by the statistician may remain slightly mysterious to the experimenter. An important contribution of a statistician is in the design of experiments and this would imply a link between the statistician and data collection. Finally, it also important for the scientist to be involved in the analysis of the data. The last diagram has added these two links and completes the posiblites for a full collaboration. This paper will argue that it is possible to foster these two nonstandard links in the final diagram throught the use of appropriate software on a personal computer. The following sections give some specific examples of how this was accomplished.

1a Short Term Consulting



1b Limited Collaboration



1c Full Collaboration

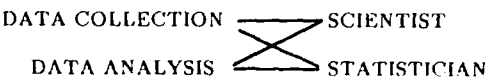


Figure 1. Some different roles for a scientist and a statistician in a reasearch project.

3.1 The overwinter survival of juvenile song sparrows

This project is based on the research by Dolph Schluter and James N.M. Smith , Department of Zoology, University of British Columbia on the song sparrow population of an island off the coast of Britsh Columbia (Schluter and Smith 1986). The interest in this project was to indentify what characteristics in a song sparrow improve its chances for surviving over the winter season. Before the winter for the years 1974 - 1979 the juvenile song sparrow population on Mandarte island was exhaustively sampled by capturing the birds in mist nets. Six morphologic measurements were made on each bird involving the size of the body and beak. The same survey was done after the winter and the birds not present on the island at that time were recorded as nonsurvivors. One way of quantifying survival is by a fitness function. If $\underline{x}$ are the six morphological measurements made on a sparrow let $p(\underline{x})$ denote the probability that this individual will survive through the winter. The functional form for p is not known and thus these ecologists feel it is important to be able to estimate p without having to assume a specific parametric

model. Note that this problem does not fit into the ordinary nonparmetric regression setting because the independent variable is a 0 or 1 response and the variance of this response depends on $p(\underline{x})$. Also, there are obvious constriants on p: $0 \le p(\underline{x}) \le 1$.

When I was first contacted by Dolph Schluter he already had an interest in analyzing these fitness data using nonparametric methods. One possible way of estimating the fitness function is by a penalized likelihood approach and the details of a spline method are described in the next subsection. The immediate problem was finding software that would run on his IBM AT. If the fitness function only depends on one independent variable then it is possible to compute p using a modest-size FORTRAN program. Dolph Schluter was able to use the program that I wrote to investigate the effect of the morphologic variables separately. Also, by having the numerical portion avaliable, he was able to spend time on a user friendly shell to call these the numerical routines. The software resulting from our combined efforts was not only statistically sound but could also be used by other ecologists with minimal introduction. The use of these nonparametric methods for fitness data has been subsequently described in Schluter (1988).

Figure 2 is an example of this nonparametric method for estimating survival. Plotted are the responses (0,1) for 151 juvenile male song sparrows against the standardized second principle component of the morphologic measurements. The smooth curves are the estimated probabilties of survival for different amounts of smoothing. The solid curve in this group is the spline estimate where the amount of smoothing was determined *objectively from the data* by cross validation.

Estimating a fitness *surface* is more complicated mainly because of the multivariate nature of the problem. One possible solution is to approximate the fitness surface using the representation from projection pursuit regression (Freidman and Steutzle, 1981). The key to this representation is the identification of linear combinations of the original measurements that give a better explanation of a song sparrow's survival. This approach is appropriate because it is reasonable to expect survival to depend on a chararteristic that is a combination of the morphological measurements. Let $f(\underline{x}) = \ln(p(\underline{x})/(1-p(\underline{x}))$ be the logit of $p(\underline{x})$. One nonparametric representation for f is

$$(3.1) \qquad f(\underline{x}) = \sum_{j=1}^{J} g_j(\underline{a}_j^T \underline{x})$$

where the vectors of coefficients are chosen so that $\underline{a}_j^T \underline{a}_j = 1$. In this model one must not only estimate the ridge functions g_j but also the projections, $\underline{a}_j$. Although this adds more structure to the statisitical method, there are computational advantages because one only needs to consider a nonparametric estimates of curves rather than an estimate of a surface.

Figure 3 gives some results for the male junvenile sparrows for two ridge functions (J=2). The amount of smoothing used in this estimate is the same as that used for the solid curve in Figure 2 ($\ln(\lambda) \approx -4$). Plotted are the probability contours of the fitness surface as a function of the two variables obtained from the two estimated projections, $\underline{a}_1$ and $\underline{a}_2$. The shape of this surface is a saddle where the first variable has a stabilizing influence on survival while the second is disruptive. One interesting feature of this estimate is that the first projection yields a linear combination of the original measurements that is similar to the second principle component. This second principle component can be identified with the relative size between the body and head.

Projection pursuit estimation especially in the context of generalized additive models is computationally intensive. The estimate plotted in Figure 3 took approximately one half an hour on a VAX 750. However it is a mistake to retreat from a method that is just beyond the power of a PC. Given the current trend for more powerful personal computers, it is largely a matter of time before these methods will be feasible. Also, the amount of time it takes to perform a statistical analysis is often evaluated on the wrong scale. The fitness data described in this section was accumulated over the course of five years at a remote site. Even if a statistical analysis takes several days on a PC this is a modest amount of time compared to the effort spent on collecting these data.

3.2 A nonparametric estimate of the fitness curve using a smoothing spline

First assume that only one morphological measurement is taken and let (y_k, x_k), $1 \le k \le n$ be the observed data where $y_k = 0$ indicates nonsurvival and $y_k = 1$ corresponds to survival of the juvenile sparrow. The log likelihood for these data is porportional to

$$\sum_{k=1}^{n} [\ln(p(x_k))y_k + \ln(1-p(x_k))(1-y_k)]$$

Now let $\phi(u) = e^u/(1+e^u)$ be the logistic link function and let $f(x) = \ln(p(x)/(1-p(x)))$. With this parametrization $p(x) = \phi(f(x))$ and has the advantage that the range of f does not have any constraints. The log likelihood now has the form:

$$\sum_{k=1}^{n} f(x_k)y_k + \psi(f(x_k))$$

with $\psi(u) = \ln(1/(1+e^u))$.

Now if this expression were maximized over all functions, f, the solution would degenerate to a function where $f(x_k) = \infty$ if $y_k = 1$ and $f(x_k) = -\infty$ when $y_k = 0$. Cleary this is not a suitable estimate. One reason that result is not appropriate is because one expects some continuity or smoothness between values of f (and p)
for similar values of x. This assumption implies that the survival of a sparrow is a continuous function of the morphological measurements. One way of incorporating this information is to penalize the likelihood when f is rough. For example, $\int f''(u)^2 du$ is one overall summary of the curvature of f and will be large when f is very wiggly and small as f becomes linear. In this work, the estimate of f is found by maximizing

$$\sum_{k=1}^{n} -f(x_k)y_k + \psi(f(x_k)) - \lambda \int (f''(u)^2 du$$

over all functions where $\int (f''(u)^2 du < \infty$. Note that although it is the logit, f, that is being estimated directly, the estimated survival probabilites are found by the transformation $\hat{p}(x) = \phi(\hat{f}(x))$.

Surprisingly this estimate is not difficult to compute and has the same form as an ordinary cubic smoothing spline. That is, the estimated curve has two continuous derivatives and can be expressed as a piecewise cubic polynomial with join points at $\{x_k\}$, $1 \le k \le n$ (see Eubank, 1988 for an introduction to this subject). Although this

maximum must be found using an iterative procedure, each iteration is efficient because it only requires the smoothing of a set of psuedo-observations using an weighted, cubic smoothing spline. This algorithm is very simlar to the iteratively reweighted least squares approach used to estimate the parameters in generalized linear models.

The smoothing parameter λ controls the relative weight given to the roughness penalty and the log likelihood. Note that when λ is very small the estimate will fit the data well but may not be very smooth. At the other extreme when λ is very large the estimate will approach a staight line where the slope and intercept are the usual maximum likelihood estimates. The effect of varying λ is shown in Figure 2.

So far the discussion has focused on computing a spline estimate for a fixed value of λ. Because the estimated curve is sensitive to the choice of this parameter, it is important to be able to estimate an appropriate value objectively from the data. One way to accomplish this is by cross validation. Let $p_{\lambda,k}$ denote the spline estimate of p for a particular value of the smoothing parameter having omitted the k^{th} observation, (y_k,x_k) . If this value for λ is a good one then "on the average" $p_{\lambda,k}(x_k)$ should be "close" to the omitted observation, y_k. This correspondence can be quantified by the cross validation function:

$$V_{cv}(\lambda) = \sum_{k=1}^{n} \frac{(y_k - p_{\lambda,k}(x_k))^2}{(1-p_{\lambda,k}(x_k))p_{\lambda,k}(x_k)}$$

where the denominator in this sum of squares adjusts for the different variances. With this criterion a data-based estimate of the smoothing parameter is the value that minimizes V_{cv} (see Yandell, et al., 1984 for more details). At first glance it may appear that V_{cv} will be very expensive to compute. However, by considering a linear approximation based on the estimate of p for the full data set and some efficient routines for cubic splines (Hutchinson and Dehoog 1985) this cross validation function can be computed easily on an IBM PC.

3.3 Projection Pursuit Estimation

In the the same manner as the univariate spline estimate, the projection pursuit estimate of the fitness surface will be defined as the maximizer of a penalized likelihood. Recall that from (3.1) $f(\underline{x})$ will depend on J pairs of univariate functions and projections. For a fixed set of projections and for some $\lambda>0$ the estimates of g_j are taken to be the functions that minimize:

$$(3.3) \qquad \sum_{k=1}^{n} f(\underline{x}_k)y_k + \psi(f(\underline{x}_k)) - \lambda \sum_{j=1}^{J} \int (g_j''(u)^2 du$$

such that $\int (g_j''(u)^2 du < \infty, \quad 1 \le j \le J.$

In this way any set of projections will determine an estimate for the surface. In ordinary projection pursuit regression, one chooses a set of projection vectors by minimizing the residual sum of squares. In this case, it is natural to consider a weighted sum of squared residuals (which will also be close to the deviance). Let $\hat{p}(\underline{x})$ denote the probability surface corresponding to the estimated logit function for a fixed set of projections. Weighting residuals by the estimated standard deviation of y_k,

$$(3.4) \qquad R(\underline{a}_1,\ldots,\underline{a}_J) = \sum_{k=1}^{n} \frac{(y_k - \hat{p}(\underline{x}_k))^2}{(1-\hat{p}(\underline{x}_k))\hat{p}(\underline{x}_k)}$$

The estimates of $\underline{a}_1,\ldots,\underline{a}_J$ are given by those vectors that that minimize R. An outline of the algorithm used to perform this minimization is:

Inititialization: $g_j=0$, $a_j=0$, $1 \le k \le J$

Repeat until convergence:
  Do l=1,J
    Fix a_j for $j \ne l$ and minimize R with respect to a_l
(*coarse search on sphere refined using the simplex method*)

Note that if only one projection is allowed to vary then the maximization to estimate g_j is just the one-dimensional spline smoothing problem described in the previous section. If this algorithm converges then the limit will be a solution to the maximization/minimization problems stated above. *When* this algorithm will converge is still an open question.

In retrospect the penalized likelihood suggests a more direct method for estimating the projections. Since the penalized likelihood in (3.3) depends both on the ridge functions and the projections it is reasonable to maximize this functional jointly over these two components. It is possible that this unified approach will be better than estimating the projections based on residuals. Ordinarily the minimization over the projections does not account for the smoothness of the implied ridge functions. As the dimension of $\underline{x}$ increases it becomes easier to find a projection that fits the data well. The limited experience from this survival data is that the estimated projections may yeild rough, possibly spurious estimates. This may be due to the fact that the minimization of R does not make any adjustment for projections that give very rough estimates of the ridge functions but never-the-less fit the data well. By considering the projections that maximize the penalized likelihood, the roughness penalty may help to control this effect.

4.1 Spatial distribution of epiphytes in the tropical canopy

One important difference between tropical and temperate forests is where nutrients are stored. Although in a temperate forest most of the nutrients are found in the soil in a tropical rain forest a significant portion of the nutrients are stored as biomass. One important component of this biomass are the epiphtyes and dead organic matter in the tree canopy. Nalini Nadkarni at the Biology Department at the University of California, Santa Barbara is interested in studying the role that the canopy plays in nutrient cycling. This research has practical implications because as rain forest is cleared for agricultural use the canopy is destroyed and thus the normal nutrient cycle is interrupted.

A first phase of this project is to quantify the architecture of trees that make up the canopy and to determine how different epiphytes are distributed throughout this region of the forest. Until recently because of its height, the canopy was inaccessible to researchers. However, by using mountain climbing equipment it is now possible to reach the canopy by ropes move safely within it. The observational data can be thought of as a three-dimensional map giving the spatial locations of branches and other features within the canopy. With this type of data one can then look for patterns in the epihpytic distribution and test for preferential sites or for competition among different species. At a more fundamental level one

can study how nutrients percolate down from the top of the canopy to the forest floor.

My collaboration in this project started with designing a method to collect canopy data. The problem is to determine the three dimensional coordinates of features in a tree without having to climb to each location of interest. Also once a method has been developed, it is important to quantify its error. Our final solution was to use the parallax veiw provided by a transit at two locations. Figure 4 is a diagram of the geometry. (The mathematical details are given in the next section.) To find the coordinates of some target in a tree, a transit is used to find the horizontal and azimuthal angles from two vantage points that are separted by a short distance (approximately three meters). The target's position is estimated by the midpoint of the shortest line segment that connects the lines of sight from the two transits. An estimate of error is the length of this segment (see Figure 4). Transforming the angular measurements at two vantage points into the xyz coordinates is too complicated to do by hand but makes for a short program on a PC. Figure 5 is a draftsman's view of a tree mapped by this procedure including the locations of two kinds of epiphytes.

Besides working out the geometry, part of my role was to provide the field assistant with software to compute the tree map coordinates at the end of a day of taking sightings. In effect I have some participation in how these data are collected since these programs can incorporate logic to spot inconsistencies or flag estimated locations that have large estimated errors. The most frustrating situation is when a bad observation is identified only after returning from Costa Rica! One use for these tree maps is to aid in subsampling a tree crown for the detailed investigation of specific branches. This is another area in which I can be involved in data collection. PC-based software can be used to guide the choice of subsamples in a manner to insure a good experimental design.

Another aspect of our collaboration is having Dr. Nadkarni (and her research assistants) particiapte in the spatial analysis of the epiphyte locations. Unlike the project in the previous section little new software is needed. Rather one needs to integrate a few special purpose functions into an existing statistical package. For example, the draftsman's view of a tree in Figure 5 was drawn using a specially written macro in S. The advantage is that most of the scientist's effort will be spent learning a standard package. This is better than having to deal with a special (and perhaps idiosyncratic) program that only performs a specific analysis of the data. Besides exploratory graphics, testing hypotheses about the spatial distributions of epiphytes can also be based in S. To do this one needs an additional S function that simulates the distribution of epiphytes on the tree network according to some null hypothesis. For example, suppose one wanted to test whether the epiphytes were uniformly distributed on the branches of a tree. One selects a test statistic that measures uniformity and calculates the value of this statistic using the coordinates of the observed data. In order to calculate the reference distribution for this statistic one simulates samples whose coordinates are uniformly distributed on the tree network and for each of these simluated samples the same test statistic is computed. By generating a a large number of samples (several hundred) one can estimate the distribution of the test statistic under the hypothesis that the epiphyte positions are uniformly distributed. To do a hypothesis test, one compares the observed value of the test statistic with the distribution determined from these simulations.

4.2 Tree mapping.

In this section a derivation is given for estimating the coordinates of a target from the direction cosines measured at two vantage points. To simplify this discussion it will be assumed that the origin is at the center of the first transit while the coordinates of the second transit are $\underline{d}=(D,0,h)$. (D is the horizontal distance between transits while h is the difference in elevation.) The horizontal and vertical angles measured by the transit are taken to have the same sense as θ and ϕ in a spherical coordinate system. Thus, if θ and ϕ are the pair of angles measured from the transit to the target then a unit vector, $\underline{e}$, in this direction has components:

$$(\cos(\theta)\sin(\phi),\ \sin(\theta)\sin(\phi),\ \cos(\phi)).$$

Let $\underline{a}$ and $\underline{b}$ denote the directions to a particular target from the two vantage points of a transit. The rays respresenting the line of sights can be parametrized by $\alpha\underline{a}$ and $\underline{d} + \beta\underline{b}$. One estimate of the target position is the midpoint of the shortest line segment joining these two rays. To find this point let $\hat{\alpha}$ and $\hat{\beta}$ be the values that minimize:

$$(4.1) \qquad \left\| \alpha\underline{a} - (\underline{d} + \beta\underline{b}) \right\|^2$$

Setting first partial derivatives equal to zero yeilds the system of equations

$$\alpha - \beta\gamma = u_1$$

$$-\alpha\gamma + \beta = -u_2$$

where $\gamma = (\underline{a})^T(\underline{b})$, $u_1 = (\underline{a})^T(\underline{d})$ and $u_2 = (\underline{b})^T(\underline{d})$.

Thus

$$\hat{\alpha} = (u_1 - \gamma u_2)/(1-\gamma^2), \quad \hat{\beta} = (\gamma u_1 - u_2)/(1-\gamma^2)$$

and the estimated target position is:

$$\hat{\alpha}\underline{a} + \left[\hat{\alpha}\underline{a} - (\underline{d} + \hat{\beta}\underline{b})\right]/2.$$

The squared length of the line segment is found by substituting $\hat{\alpha}$ and $\hat{\beta}$ into (4.1).

As a final note, care should be taken in interpreting the line segment length as an absolute measure of the estimated position's accuracy. It is a biased estimate of distance between the estimated position and the actual one. Simluations indicate that in the situations encountered in mapping tree positions the median segment length is typically about 2/3 the actual distance.

References

Eubank,R. (1988). *Spline smoothing and nonparametric regression*. Marcel Dekker, New York.

Friedman J. H. and Stuetzle, W. (1981) Projection pursuit regression. *J. Amer. Statist. Assoc.* 76 817-823.

Hastie, T. and Tibishirani, R. (1986). Generalized additive models. *Statistical Science* 1 297-318.

Hutchinson, M.F. and de Hoog, F.R.(1985). Smoothing noisey data with spline functions. *Numerische Mathematika* 47. 107-112.

References (cont.)

O'Sullivan, F., Yandell, B. and Raynor, W. (1984) Automatic smoothing of regression functions in generalized linear models. *J. Amer. Statist. Assoc.* .

Schluter,D. and Smith, J. N. M. (1986). Natural selection on beak and body size in the song sparrow. *Evolution* 40 221-231.

Acknowledgements

I would like to thank Dolph Schluter and James Smith for generously making the sparrow data availble. VAX computing was provided by a National Science Foundation Equipment grant while the work on tree mapping was supported in part by the Whitehall Foundation and National Science Foundation grant BSR86-14935.

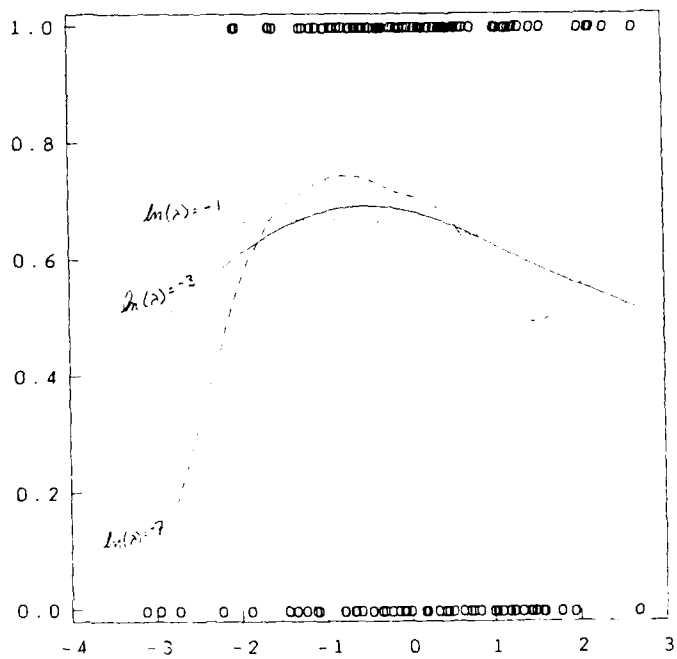


Figure 2. Over winter survival of male juvenile song sparrows as a function of the standardized second principle component. Plotted points are the actually survival of 151 birds (0= nonsurvival, 1= survival). The curves are the estimated probabilites for survival for different amounts of smoothing. In the solid curve the smoothing pararmeter has been chosen objectively from the data using cross validation.

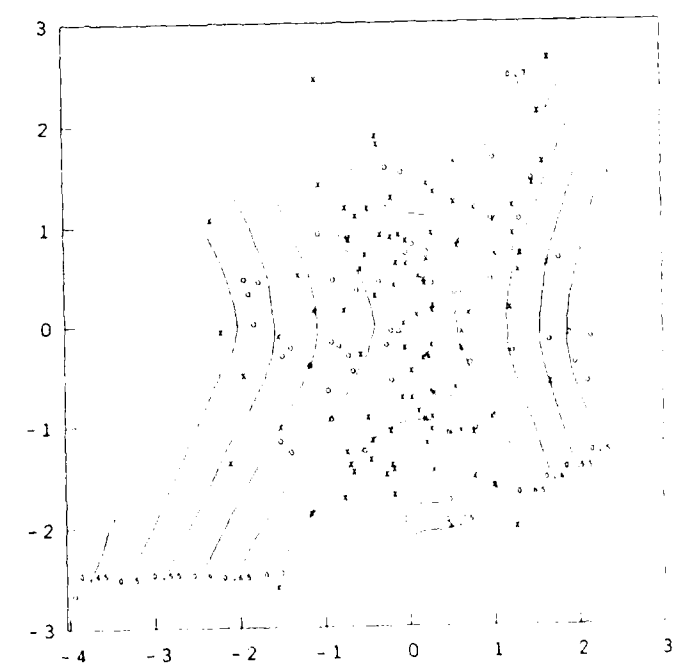


Figure 3. Over winter survival surface estimated using projection pursuit $\hat{\Lambda}$ function of two projections of the morphologic measurements. Plotted are the probability contours for a surface consisting of two ridge functions. The first projection (x axis) is very similar to the second principle component described in Figure 2.

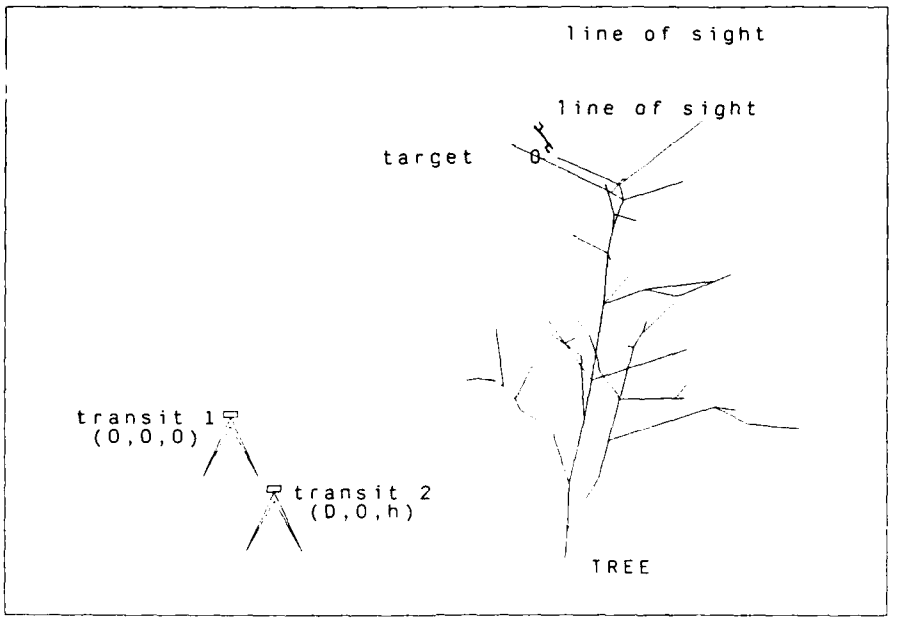


Figure 4. Geometry of measuring position from the angular measurements of two transits.

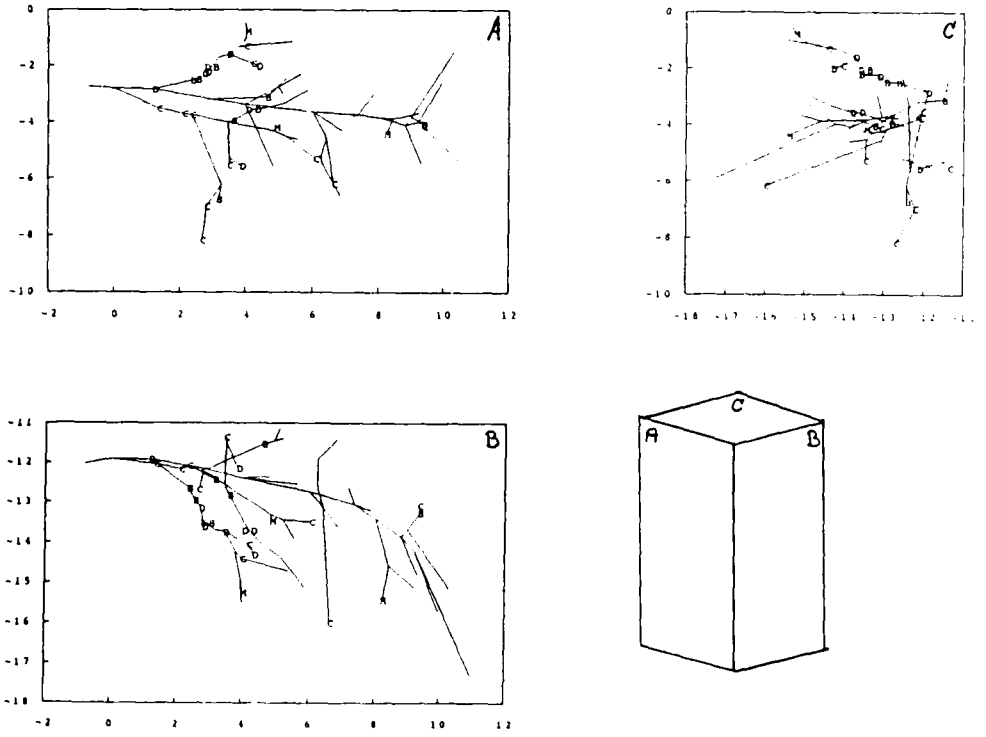


Figure 5. Tree map. Draftsman view (top and two side views at right angles) of a pasture tree from a study site in Costa Rica. The scale is in meters and the plotted symbols indicate the location of bromeliads (B-D) and mistletoes (M).

SPACE BALLS!
OR
ESTIMATING THE DIAMETER DISTRIBUTION OF MONOSIZE POLYSTYRENE MICROSPHERES

Susannah B. Schiller, National Bureau of Standards

Introduction.

Polystyrene microspheres, with nominal diameters in the range of 0.3 to 30 microns, have been certified by the National Bureau of Standards as Standard Reference Materials. Some of them were manufactured in space on the shuttle Challenger because the beads are more uniform in size and shape when made in zero gravity. They provide an important tool for calibrating instruments that are used to examine very small particles, such as blood cells, bacteria, or airborne dust. In order to be useful for calibration, their diameters must be well-characterized.

To certify these SRMs, the beads were put into a suspension which, when dried, caused the beads to form chains on a microscope slide. Parallel light, projected up through the slide, marked the center of each sphere in the common back-focal plane on which the microscope was focused. Because the "chained" spheres touched, the distance between sphere centers on photomicrographs gave a good estimate of sphere diameter [1]. In order to get the desired accuracy for certification, the scientists had to make careful and tedious measurements on thousands of pairs of spheres. For the users of these SRMs, who only want to verify that their mean measurements fall within the certified bounds of uncertainty, a quicker approach is desirable.

The proposed technique uses closely packed hexagonal arrays of the microspheres instead of chains. Row lengths are measured between the centers of the end spheres. The obvious diameter estimate is the average center-to-center distance, found by dividing the row length by the number of spheres in the row minus one. However, because the diameters are not identical, there are always air gaps in these arrays which inflate the diameter estimates. These air gaps cannot be measured via the center distance finding technique, nor have they been modelled mathematically. Additionally, there is the problem of the "scrunching factor," or Van der Waals' attractions. When two objects touch, they flatten by some factor which, in the case of polystyrene microspheres of the size under consideration here, is about 0.1% . This factor was easily taken into account for the pairwise measurements of microspheres arranged in chains, where it was known that every pair of spheres touched. However, in an array where many pairs of spheres do not touch, the analysis is much more difficult.

Simulation.

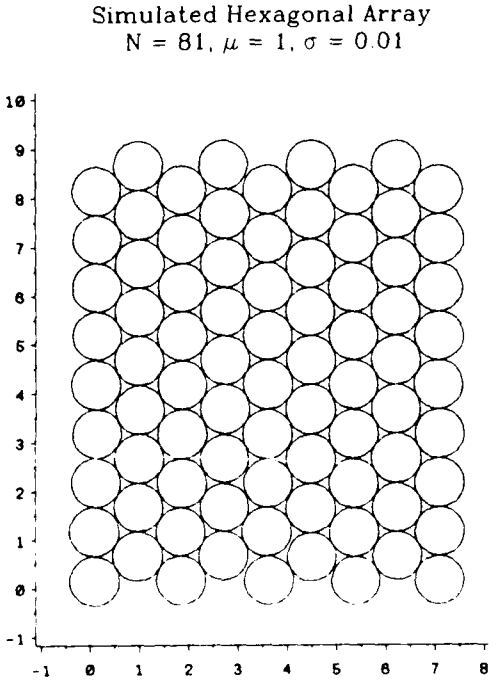
The approach taken to this estimation problem was to simulate packed arrays of spheres and determine the behavior of the air gaps. From the chain measurements, it was known that the diameters followed a normal distribution, and that the standard deviations were roughly 1% of the mean diameter. To simulate this, arrays of circles whose diameters came from the normal distribution $N(\mu_D, \sigma^2)$ were generated. These were "packed" by minimizing the sum of squared distances between centers of neighboring circles subject to the following constraint: if packing caused a pair to touch, they were forced to overlap by exactly 0.1% of the average of the two diameters involved. Otherwise, an air gap was left whenever the centers were more than the average of the two diameters apart.

Multiple simulations were performed for each of the following combinations of N, μ_D and σ, where N is the number of circles in an array:

N = 81 μ_D = 1 σ = 0.009 to 0.015 by 0.001
N = 64 μ_D = 1 σ = 0.008 to 0.015 by 0.001
N = 64 μ_D= 1.5 σ = 0.008μ_D to 0.015μ_D by 0.001μ_D
N = 64 μ_D = 2 σ = 0.009μ_D to 0.012μ_D by 0.001μ_D

Each array was laid out in a "square" fashion, with $\sqrt{N} = K$ columns and K circles per column (Figure 1).

Figure 1



The arrays were packed by minimizing the sum (over the entire array) of the squared gaps between neighboring circles, subject to not allowing the circles to overlap, using the routine E04VCF in NAG. Larger arrays caused this routine to fail, and smaller arrays were deemed too small to be useful. Results were output in the form of center coordinates and a diameter for each circle in the array. The distance between each neighboring pair of circles was found, and row lengths R (between the centers of the outer circles of each row) were measured in all three possible directions. Only rows with a fixed number of circles, K, were considered for either the row lengths or pairwise distances. Average center-to-center distances were computed from the row lengths:

$$C = \frac{R}{K-1}$$

and were averaged for each array.

Gap Frequency and Size Distribution.

A colleague has conjectured that, assuming the variability among diameters is "small," the minimum proportion of gaps possible in an array of circles is the number of interior circles divided by the total number of neighboring pairs. For the case of the "square" array, this is:

$$p_m = \frac{(K-2)^2}{(K-1)\ (3K\ 1)} \qquad (1)$$

This means that the minimum percent of gaps for the simulated arrays should be:

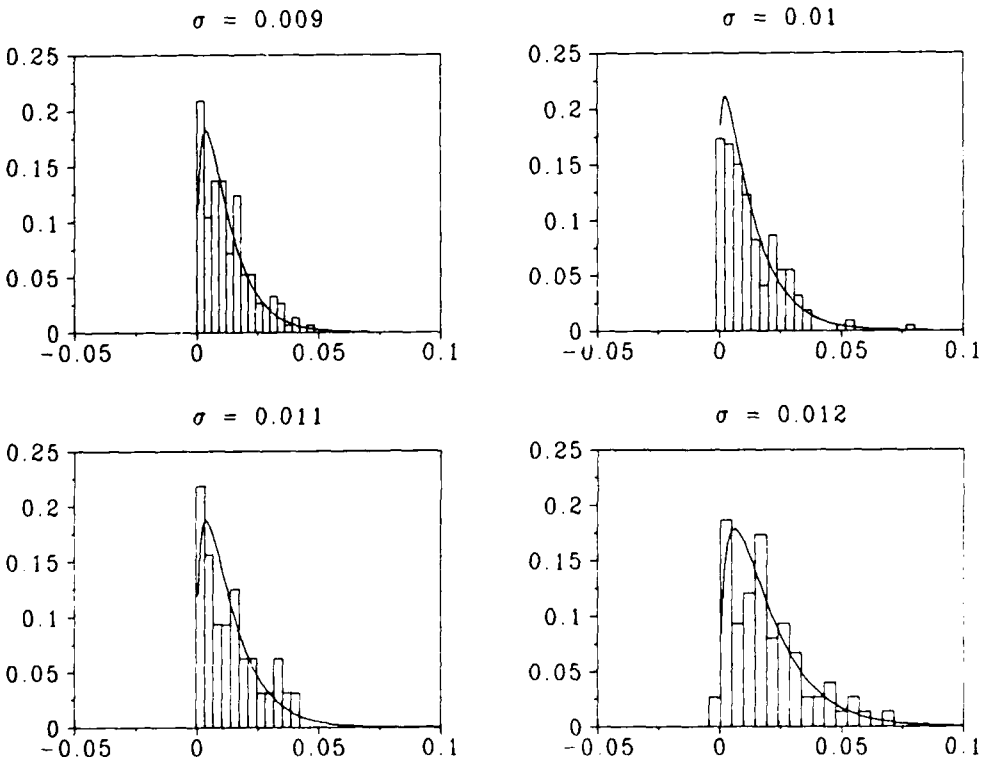
K	Percent Gaps
8	22.36%
9	23.56%

The conjecture serves as a useful guide. For all relative standard deviations considered, an air gap existed between a pair of circles about 25% of the time in the simulated arrays. It might be noted, however, that this minimum proportion can be greater than 25% for larger arrays. For example, a 14 x 14 array should have gaps between neighboring circles at least 27% of the time, based on the conjecture.

The gaps appear to follow a gamma distribution. Histograms of the gaps were overlaid with plots of gamma probability density functions having the same mean and variance, and the fit was remarkably good (Figure 2). To date, no theoretical reason has been determined for this occurrence.

Figure 2

Fit of Gamma Pdf to Histogram of Gaps



The mean and standard deviation of the gaps depend upon the standard deviations of the circle diameters, but these statistics are quite variable between simulated arrays. This is to be expected, because gap sizes depend on the overall layout of the array, not just on the two diameters on either side. For example, if the same N balls were arranged differently in the square array, the optimization would produce a totally different set of gaps. Bearing in mind this variability, we found empirically that both the gap average and standard deviation can be approximated as multiples of the diameter standard deviation (Figures 3 and 4):

$$\bar{G} \approx 1.3443\, S_D \qquad (2)$$

$$S_G \approx 1.1277\, S_D \qquad (3)$$

Figure 3

Average Gap = 1.3443 * Diameter Standard Deviation

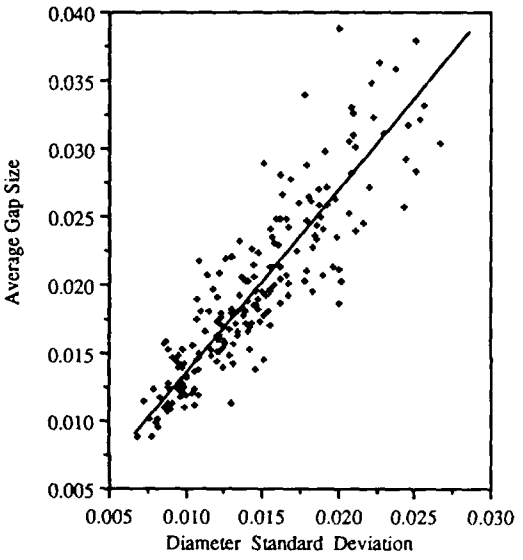
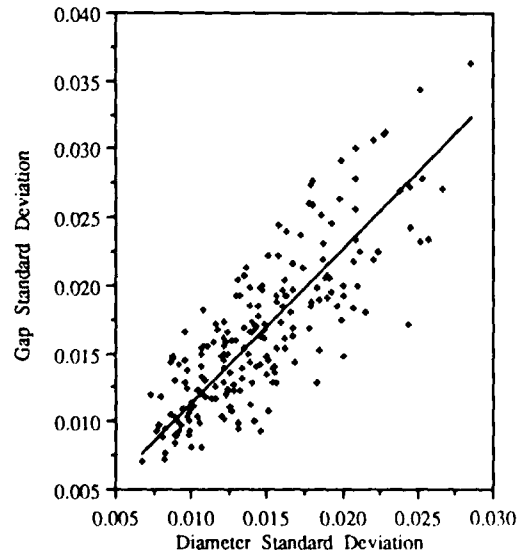


Figure 4

Gap Std = 1.1277 * Diameter Standard Deviation



Models for Center-to-Center Distance Mean and Variance.

Using information about the average frequency with which gaps occur and their size distribution, functional relationships between the diameter mean and standard deviation and the array center-to-center distance mean and standard deviation can be found.

A full model for the average center-to-center distance, C, for a row of K balls, is:

$$C = \frac{1}{K-1}\sum_{1}^{K-1}\left\{\frac{D_i+D_{i+1}}{2} + Z_iG_i - 0.001(1-Z_i)\frac{D_i+D_{i+1}}{2}\right\} \qquad (4)$$

where:

$D_i \sim N(\mu_D, \sigma^2)$ the circle diameters

$G_i \sim \Gamma(\alpha,\beta)$ the gaps ($\mu_G = \alpha\beta$, $\sigma_G^2 = \alpha\beta^2$)

$Z_i \sim B(1,p)$ a binomial random variable denoting whether or not a gap occurs

This leads to a very messy computation for the variance, especially if all of the possible covariances between random variables are considered. There is some correlation between the D_i and G_i and Z_i , respectively, but it is small and will be disregarded. A further simplification is to ignore the fact that the Z_i are random variables, and replace the Z_i in the formula by their expectation, p:

$$C = \frac{1}{K-1}\sum_{1}^{K-1}\left\{\frac{D_i+D_{i+1}}{2} + pG_i - 0.001(1-p)\frac{D_i+D_{i+1}}{2}\right\} \qquad (5)$$

This gives

$$E[C] = (0.999 + 0.001p)\mu_D + p\mu_G$$

and

$$\mathrm{Var}(C) = \frac{1}{K-1}\left\{\frac{(2K-3)(0.999 + 0.001p)^2}{2(K-1)}\sigma^2 + p^2\sigma_G^2\right\}$$

From this model, the natural estimators are:

$$\hat{C} = (0.999 + 0.001p)\bar{D} + p\bar{G} \qquad (6)$$

and

$$\hat{\mathrm{Var}}(C) = \frac{1}{K-1}\left\{\frac{(2K-3)(0.999 + 0.001p)^2}{2(K-1)}S_D^2 + p^2S_G^2\right\} \qquad (7)$$

Applying equations (2) and (3) to equations (6) and (7) yields:

$$\hat{C} \approx (0.999 + 0.001p)\bar{D} + 1.3443p\, S_D \qquad (8)$$

and

$$\hat{\mathrm{Var}}(C) \approx \frac{1}{K-1}\left\{\left(\frac{(2K-3)(0.999 + 0.001p)^2}{2(K-1)} + (1.1277p)^2\right)S_D^2\right\} \qquad (9)$$

We can also estimate $\hat{C}$ and $\hat{\mathrm{Var}}$ (C) by

$$\bar{C} = \frac{1}{M}\sum_{1}^{M} C_j$$

and

$$S_{\bar{C}}^2 = \frac{1}{M-1}\sum_{1}^{M}(C_j - \bar{C})^2$$

where M rows have been measured and divided by K-1 to produce the Cj.

Predicted Diameter Standard Deviation.

Of course, the real interest is in finding estimates of μ_D and σ in terms of $\bar{C}$ and S_C. Eq. 9 suggests that a model for σ should look like:

$$\sigma = a\sqrt{(K-1)Var(C)}$$

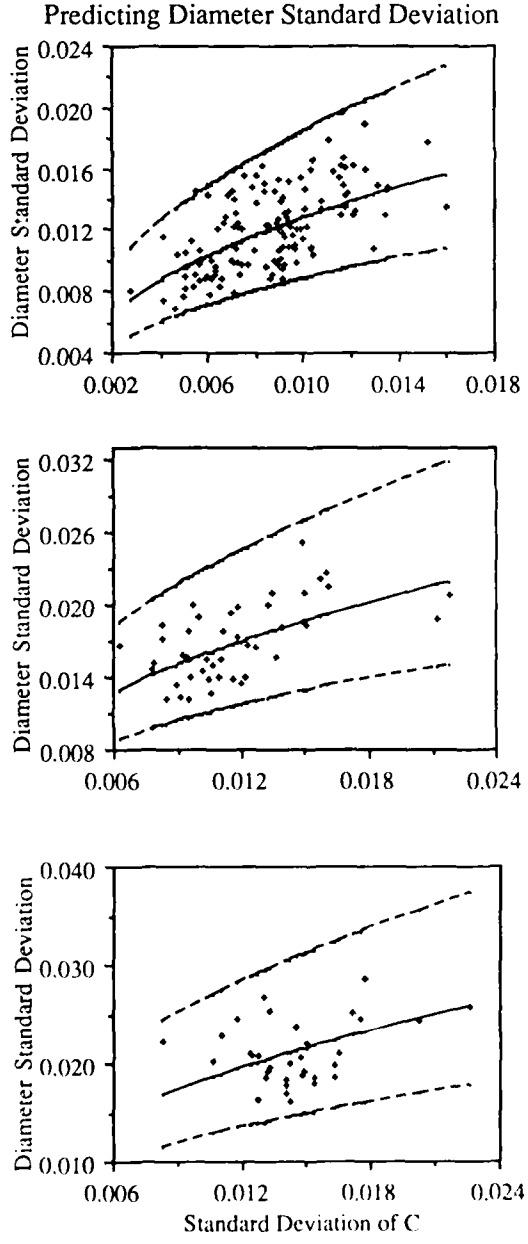
However, we assume that the logarithm of σ is more nearly normally distributed than σ itself, so this fit is better done on the log scale:

$$\ln(\sigma) = a\ln(\sqrt{(K-1)Var(C)}) + b$$

and experimentation showed that a much better fit is found when $\ln(\bar{C})$ is included in the model.:

$$\ln(\sigma) = a\ln(\sqrt{(K-1)Var(C)}) + b\ln(\bar{C}) + c \quad (10)$$

Figure 5



Estimators S_D and S_C from the simulation data were used for σ and $\sqrt{Var(C)}$ when fitting this model Using all of the data from the described simulations, the parameter estimates from this fit are $\hat{a} = 0.4282$, $\hat{b} = 0.5061$ and $\hat{c} = -2.3913$. To find a 95% prediction interval for σ we first propagate the errors in $\bar{C}$, S_C, and the parameter estimates $\hat{a}$, $\hat{b}$ and $\hat{c}$ to estimate the variance in $\ln(\hat{\sigma})$ due to uncertainty in the model.

Variance Due to Model =

$$MSE\ (x_*'\ (X'X)^{-1}x_*) + \hat{b}^2\frac{S_C^2}{M\bar{C}^2} + \frac{\hat{a}^2}{2(M-1)}$$

where X is the design matrix, MSE is the mean square error from the fit of the linear model, and

$$x_*' = (\ \ln(\sqrt{(K-1)}S_C)\quad \ln(\bar{C})\quad 1\ ).$$

The total variance of $\ln(\hat{\sigma})$ is:

$$\hat{Var}(\ln(\hat{\sigma})) = MSE\ (x_*'\ (X'X)^{-1}x_* + 1) + \hat{b}^2\frac{S_C^2}{M\bar{C}^2} + \frac{\hat{a}^2}{2(M-1)}$$

Assuming that enough measurements (M) were made so that two standard deviations is the appropriate width, the 95% prediction interval for $\ln(\sigma)$ is given by

$$\ln(\hat{\sigma}) \pm 2\sqrt{\hat{Var}(\ln(\hat{\sigma}))}$$

and the 95% prediction interval for σ is:

$$\left(\exp(\ln(\hat{\sigma}) - 2\sqrt{\hat{Var}(\ln(\hat{\sigma}))}\ ),\ \exp(\ln(\hat{\sigma}) + 2\sqrt{\hat{Var}(\ln(\hat{\sigma}))}\ )\right)$$

Figure 5 shows the fitted data and prediction limits (all on original scale) plotted against $\sqrt{(K-1)}S_C$. The plot has been broken into three sections for the three nominal values of μ_D that were used to generate the simulation data.

Predicted Diameter Mean.

The estimator for $\hat{C}$ suggests a fit for μ_D. Equating Eq. 8 to $\bar{C}$ gives a function for $\bar{D}$ in terms of $\bar{C}$, S_D, and p:

$$\bar{D} \approx \frac{\bar{C}}{0.999+0.001p} - \frac{1.3443pS_D}{0.999+0.001p}$$

Applying the fitted Eq. 10 gives an estimator for $\hat{\mu}_D$ in terms of $\bar{C}$, p, and $\sqrt{K-1}S_C$:

$$\hat{\mu}_D = \bar{D} \approx \frac{\bar{C}}{0.999+0.001p} - \frac{1.3443p}{0.999+0.001p}(0.0915\,\bar{C}^{0.5}(\sqrt{K-1}S_C)^{0.4}) \quad (11)$$

When a function of this form was fit to the simulation data directly, it was found that the power to which $\bar{C}$ was raised went to 0. Thus, a function of the form:

$$\bar{D} = \frac{a'\bar{C}}{(0.999+0.001p)} - \frac{b'p\,(\sqrt{K-1}S_C)^{c'}}{(0.999+0.001p)}$$

is reasonable. However, the residual sum of squares was virtually the same for this fit as for the simpler linear model:

$$D = \frac{(1-a'p)\bar{C}}{(0.999+0.001p)} - \frac{b'p\sqrt{K-1}S_C}{(0.999+0.001p)} \quad (12)$$

so the latter model was applied.

Unfortunately, p cannot be estimated from photomicrographs, so its lower bound, determined by the number of circles in each row measured (Eq. 1), is used. The parameter estimates from fitting Eq. 12 with the simulation data are $\hat{a}' = 0.0091$ and $\hat{b}' = 0.9736$. To find a 95% prediction interval for μ_D we again propagate the errors in $\bar{C}$, S_C, and the parameter estimates $\hat{a}'$ and $\hat{b}'$ to estimate the variance in $\hat{\mu}_D$ due to uncertainty in the model:

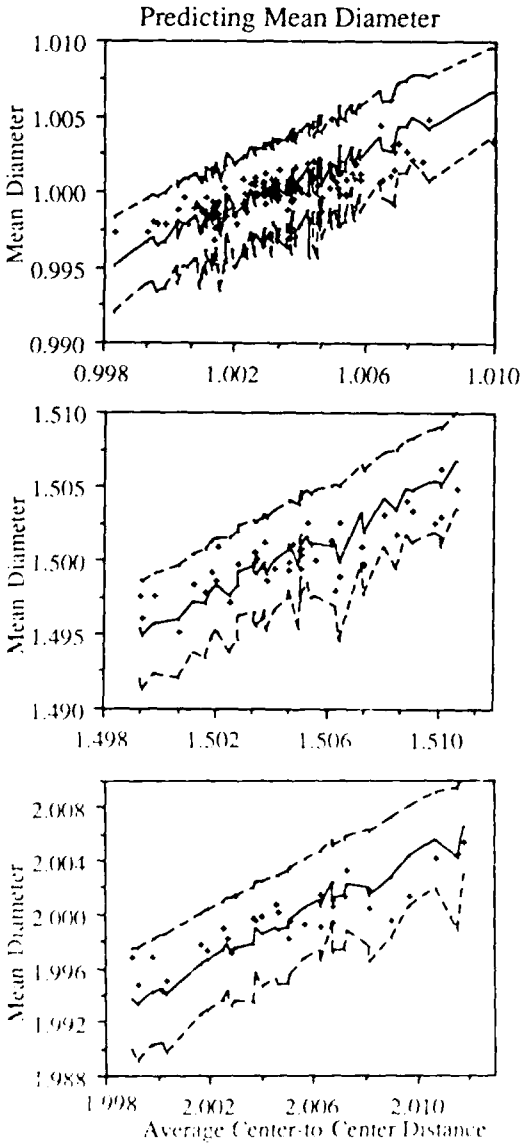
Variance Due to Model =

$$\frac{MSE\left(x_*'(X'X)^{-1}x_*\right)+\frac{S_C^2}{M}(1-\hat{a}'p)^2+\frac{\hat{b}'^2p^2(K-1)S_C^2}{2(M-1)}}{(0.999+0.001p)^2}$$

where X is the design matrix, MSE is the mean square error from the fit of the linear model, and

$$x_*' = \left(\ \bar{C}\ \sqrt{K-1}S_C\ \right)$$

Figure 6



Thus, the total variance, V, for $\hat{\mu}$ is:

$$\frac{MSE\left(x_*'(X'X)^{-1}x_*+1\right)+\frac{S_C^2}{M}(1-\hat{a}'p)^2+\frac{\hat{b}'^2p^2(K-1)S_C^2}{2(M-1)}}{(0.999+0.001p)^2}$$

Assuming that enough measurements (M) were made so that two standard deviations is the appropriate width, the 95% prediction interval for μ is given by

$$\hat{\mu} \pm 2\sqrt{V}$$

Figure 6 shows the fitted data and prediction limits plotted against $\bar{C}$. It is broken into three sections for the three nominal values of μ_D that were used to generate the simulation data. A curious artifact of the data, which has not been explained to date, is that the slope of the regression line for the entire data set is different than the slope for any of the three subsets.

Empirical Results.

The estimates for $\hat{\mu}_D$ and $\hat{\sigma}$ were tested on a small amount of real data from arrays of (nominally) 3 micron polystyrene microspheres. The diameters of these microspheres have a certified mean of 2.978 microns and a certified standard deviation of 0.025 microns. The results from three packed arrays, after corrections were made for random and systematic error due to the photographic and measurement processes, are (in microns):

	$\bar{C}$	S_C	K	$\hat{\sigma}$	$\hat{\mu}_D$
Array 1	2.9826	0.0068	14	0.033	2.971
Array 2	2.9812	0.0058	11	0.029	2.972
Array 3	2.9835	0.0096	17	0.040	2.967

	95% Prediction Limits for σ		95% Prediction Limits for μ_D	
Array 1	0.023	0.047	2.962	2.980
Array 2	0.020	0.042	2.963	2.980
Array 3	0.027	0.057	2.958	2.977

The predictions $\hat{\mu}_D$ are consistently lower than the certified value, suggesting that the simulation is not packing the circles as tightly as the spheres are packed in reality. However, the 95% prediction intervals do cover or nearly cover the certified value. Also, the prediction intervals for σ are narrow enough that they will be useful.

Previous Study of Hexagonal Arrays.

The idea of measuring row lengths in hexagonal arrays to glean information about the mean diameter is not new. The relative size of the bias introduced by air gaps was studied empirically by Kubitschek in 1961 [2], using arrays of 100 washers. Results here suggest that his estimate of bias is too large for this application, as will be shown below.

Kubitschek found that

$$C = D + 0.36\,S_D \qquad (13)$$

(Note that S_D is not available for the center distance finding technique which is why the estimation above did not use it). To account for the flattening that takes place

when two polystyrene microspheres touch, $\bar{C}$ is divided by (0.999+0.001p) for the present application, giving:

$$\frac{\bar{C}}{(0.999 + 0.001p)} = \bar{D} + 0.46\, S_D.$$

From the model

$$\bar{D} = \frac{\bar{C}}{0.999 + 0.001p} - \frac{p\bar{G}}{0.999 + 0.001p}$$

and Eq. 2 we get

$$\bar{D} = \frac{\bar{C}}{0.999 + 0.001p} - \frac{1.3443pS_D}{0.999 + 0.001p} \qquad (14)$$

This suggests doing a direct fit of

$$\frac{\bar{C}}{0.999 + 0.001p} = \bar{D} + apS_D \qquad (15)$$

using the lower bound for p ($\hat{a}$ = 1.4889). Both Eq. 14 and Eq. 15 give much smaller estimates of the coefficient of S_D than Kubitschek gave:

	Coefficient of S_D	
Method	N = 64	N = 81
Model (Eq 14)	0.30	0.32
Direct Fit (Eq. 15)	0.33	0.35

Perhaps Kubitschek's overestimate can be explained by the fact that the majority of his data is from washer distributions with just two or three distinct sizes. It is interesting to note that, applying Kubitschek's equation to the empirical data, we would get more dramatic underestimates of the certified value than we do using the results from the simulation.

Conclusion.

Standard Reference Materials of polystyrene microspheres are used for calibrating optical microscopes. Packing the spheres into hexagonal arrays instead of forming chains with them on a microscope slide gives a relatively quick measurement technique for doing this. However, the raw estimates of diameter mean and standard deviation are biased because of air gaps between some pairs of spheres. So far, simulation has proven to be the only way to examine this bias and develop a correction for it.

Acknowledgements:

I would like to gratefully acknowledge A.W. Hartman of the Precision Engineering Division of NBS for presenting the problem, supplying measurement data with which to compare the results, and providing helpful discussions on the physics of the particle sizing process.

I would also like to acknowledge K. R. Eberhardt of the Statistical Engineering Division of NBS for his insight and discussions about modelling the simulation data.

Finally, I would like to acknowledge Christoph Witzgall and Jim Lawrence for their conjecture on the minimum number of gaps possible in an array of circles whose diameters differ only slightly.

References:

[1] A.W. Hartman, Powder Technology, 46 (1986) 109.

[2] H.E. Kubitschek, Nature, 192 (1961) 48.

Maximum Queue Size and Hashing with Lazy Deletion

(extended abstract)

Claire M. Mathieu, Princeton University

Jeffrey Scott Vitter, Brown University

1. Introduction

Queueing phenomena are widespread in the fields of operating systems, distributed systems, and performance evaluation. Queues are also a natural way to model the size of classical dynamic data structures, such as buffers, dictionaries, sets, stacks, queues, priority queues, and sweepline structures. As a consequence, many statistical properties of queues have been investigated, such as their expected size and variance. Yet, very little was known about the *maximum* size of queues over a given period of time. If the size of the queue represents the amount of resource used by a computer program or a systems component, then such information is important for making intelligent decisions about preallocating resources.

Another motivation for our study was the need to develop and analyze practical space-efficient methods for processing sweepline information. Some work in this area has been done by Vitter and Van Wyk [1986], Morrison, Shepp, and Van Wyk [1987], and Ottmann and Wood [1986], but as the latter point out, "Surprisingly there has been little theoretical investigation of space-economical plane-sweep algorithms even though such algorithms have significant practical applications." Ottmann and Wood [1986] do not investigate the maximum number of items cut by the sweepline; they express the running times of their algorithms in terms of the maximum number. Our approach in this paper is to examine the distribution of the maximum number of cut items, based on several popular input models, and in addition show that the "hashing with lazy deletion" (HwLD) algorithm introduced in [Van Wyk and Vitter, 1986] is extremely practical and optimum in both average running time and preallocated space.

We develop new methods and obtain several results about the distribution of the maximum queue size, under several *models of growth*. We study stationary birth and death processes, and are particularly interested in M/M/∞ and the more general M/G/∞ queues, which model the amount of plane sweep information as a function of time. We also concentrate on HwLD, which is a non-Markovian queueing model corresponding to the space usage of the algorithm by the same name. In addition we study a non-stationary model corresponding to histories of priority queues.

Plane sweep algorithms process a sequence of items over time; at time t the data structure stores the items that are "living" at time t. Let us think of the ith item as being an interval $[s_i, t_i]$ in the unit interval, containing a unique key k_i of supplementary information. The ith item is "born" at time s_i, "dies" at time t_i, and is "living" at time t when $t \in [s_i, t_i]$. The data structure must be able to support the dynamic operation of searching the living items based on key value. It is natural to think of the data structure as a queue, as far as size is concerned. Let us denote the queue size at time t by $Need(t)$, the number of items that need to be included in the data structure. If we think of the items as horizontal intervals, then $Need(t)$ is just the number of intervals "cut" by the vertical sweepline at position t. In a typical application, we may have 10^6 intervals in the time range $[0, 1]$, with $E(Need) = 10^3$; that is, only square root of the total number of items tend to be present at any given time [Szymanski and Van Wyk, 1983]. It is thus very inefficient to devote a separate storage location to every item; the data structure should be dynamic.

In HwLD, items are stored in a hash table of H buckets, based upon the hash value of the key. The distinguishing feature of HwLD is that an item is not deleted as soon as it dies; the "lazy deletion" strategy deletes a dead item only when a later insertion accesses the same bucket. The number H of buckets is chosen so that the expected number of items per bucket is small. HwLD is thus more time-efficient than doing "vigilant deletion," at a cost of storing some dead items.

Let $Use(t)$ be the number of items in the HwLD data structure at time t. It is shown in [Van Wyk and Vitter, 1986] for the M/M/∞ model that at any given time t we have

$$E(Use) = E(Need) + H = \frac{\lambda}{\mu} + H,$$

where λ is the birth rate of the intervals and $1/\mu$ is the average lifetime per item. The amount of wasted space is equal to the number H of buckets. A possible choice of H is $H = \Theta(E(Need))$, so that the expected amounts of space and time used by HwLD are optimal, up to a constant factor. (In practice, the computer memory space used by HwLD is often *less than* the space used by "vigilant deletion" strategies, because the latter are typically based on balanced trees and priority queues, which require more storage overhead (pointer information) per item.) It was conjectured in [Van Wyk and Vitter, 1986] that

$$E(\max_{0 \le t \le 1}\{Need(t)\}) \quad O\left(\frac{\lambda}{\mu}\right)$$

$$E(\max_{0 \le t \le 1}\{Use(t)\}) \quad \max_{0 \le t \le 1}\{Need(t)\} \quad O(H),$$

which would prove that HwLD is also optimal in terms of *preallocated* storage. A system of equations for the distribution of $\max_t\{Need(t)\}$ and for the degenerate

$H = 1$ distribution of $\max_t\{Use(t)\}$ in equilibrium for the M/M/∞ model was recently developed in [Morrison, Shepp, and Van Wyk, 1987]. They can be used to get numerical data. Both distributions are nearly identical, because when $H = 1$ we have $\max_{t>t^*}\{Need(t)\} = \max_{t>t^*}\{Use(t)\}$, where t^* is the birthtime of the first item to enter the queue after time $t = 0$.

In this paper we attain an array of results about the maximum queue size using two independent approaches. (Due to space limitations, details are deferred to the full paper.) In the first approach, described in the next section, we develop several formulas for the distribution of $\max_t\{Need(t)\}$ for general birth-and-death processes (which includes the M/M/∞ process) and for the distribution of $\max_t\{Use(t)\}$ in the general $H > 1$ case of HwLD. We also handle a non-stationary model described in [Vitter and Van Wyk, 1986]. The formulas provide exact numerical data on the distributions, and in some cases lead to asymptotics as the time interval grows. There is a common underlying structure in the formulas for the different models: the transform of interest in each case is the ratio of consecutive classical orthogonal polynomials.

In our second approach, described in Section 3, we prove the above conjectures for the general M/G/∞ model, which includes M/M/∞ as a special case. We obtain optimal big-oh bounds on the expected maximum queue size by using non-queueing theory techniques. We approximate the continuous time processes $\max_t\{Need(t)\}$ and $\max_t\{Use(t)\}$ by sums of discrete quantities related to hashing, specifically, maximum slot occupancies. (The hashing in our approximation scheme has nothing to do with the hashing inherent in HwLD.) Our techniques also seem applicable to other queueing models, such as M/M/1.

2. Formulas for Maximum Queue Size

It is convenient to extend the range of time to $[0, T]$ for arbitrary T; the results can be translated back to $T = 1$ later. In the following sections we derive exact formulas for the distribution of the maximum queue size in several models. Our formulas are amenable to numerical calculation and yield asymptotic expressions in some cases.

The problem has been studied previously in [Morrison, Shepp, and Van Wyk, 1987] for the special cases of M/M/∞ and the $H = 1$ case of HwLD. However, analysis for the case $H = 1$ cannot be used to get a good bound for when $H > 1$; a corollary of our analysis in Section 3 is that $H \max_t\{Use_1(t)\}$ is typically greater than $\max_t\{Use(t)\}$ by more than a constant factor, where $Use_1(t)$ is the occupancy of bucket 1 at time t.

A birth and death process is a Markov process in which transitions from level k are allowed only to levels $k+1$ and $k-1$. We shall restrict ourselves to continuous time in this exposition. Borrowing notation from HwLD, we define $Need(t)$ to be the level of the process at time t. The infinitesimal birth and death rates at level k are denoted λ_k and μ_k.

For the special case of the M/M/∞ model, we write $\lambda_0 = \lambda_1 = \cdots = \lambda$ and $\mu_k = k\mu$. For the M/M/1 model, we write $\lambda_0 = \lambda_1 = \cdots = \lambda$ and $\mu_1 = \mu_2 = \cdots = \mu$. In both cases, the arrival process is Poisson, and for the M/M/∞ case the lifespans are exponentially distributed. The reader can consult [Kleinrock, 1975] for further background.

In Sections 2.1–2.5, we derive exact formulas for the maximum queue size using a variety of algebraic and analytical techniques. The first three sections handle the case of general homogeneous and stationary birth-and-death processes in equilibrium at $t = 0$, the fourth discusses HwLD under the M/M/∞ model in equilibrium at $t = 0$, and the last deals with a non-stationary model.

2.1. Applications of Stack Histories

A Dyck path is a walk in $\mathbf{Z}^2$ above the x-axis such that each step is of the type $(a, b) \to (a+1, b \pm 1)$. Its level is the maximal y-coordinate reached. Dyck paths are a special case of file histories: they correspond to histories of stacks [Flajolet, Françon, and Vuillemin, 1980]. (File histories will be discussed further in Section 2.3.) Let ω be a Dyck path going from level r to level ℓ in n steps, and with height constrained to be $\le k$. For each such ω, we define $p_\omega(T)$ to be the probability that in time interval $[0, T]$ the successive different states of the process $Need(t)$ correspond exactly to ω, given that $Need(0) = r$.

Lemma 2.1. *We have*

$$\Pr\{\max_{0\le t\le T}\{Need(t)\} \le k\} = \sum_{r,\ell,n,\omega}\Big(\Pr\{Need(0) = r\}\cdot p_\omega(T)\Big).$$

As an example of our method, let us consider the M/M/1 model with parameters λ and μ. The equilibrium probabilities are given by $\Pr\{Need = i\} = (\lambda/\mu)^i(1-\lambda/\mu)$. It remains to calculate the $p_\omega(T)$ terms, which can be expressed as a multiple integral. In fact, $p_\omega(T)$ does not depend upon the actual shape of ω, but only upon the number of times the path hits the x-axis. Using that gives us $p_\omega(T)$ in simple summation form. Lemma 2.1 can thus be applied to yield an exact expression for $\Pr\{\max_{0\le t\le T}\{Need(t)\} \le k\}$.

2.2. Orthogonal Polynomials

We can extend the approach used by [Morrison, Shepp, and Van Wyk, 1987] for the M/M/∞ model to general birth and death processes. We have

$$\Pr\{\max_{0\le t\le T}\{Need(t)\} \le k\} = \sum_{r,\ell\le k}\Pr\{Need(0) = r\}\int_0^T s_{r,k}(t)\,dt,$$

where $s_{r,k}(t)$ is the density of the first passage time to level k starting from level r.

These densities $\{s_{r,k}\}_{r,k}$ are solutions of a system of integral equations; taking Laplace transforms $s_{r,k}(\cdot) \to$

get another system, from which we find that $\sigma_{j,k}(s) = \omega_j(s)/\omega_k(s)$ is a rational fraction; its poles are roots of ω_k, and yield $s_{j,k}(t)$ and thus $\Pr\{\max_{0\le t\le T}\{Need(t)\} \le k\}$. Moreover, when $Need(t)$ is a birth-and-death process, computing the roots of ω_k is an easier task because $\{\omega_j\}$ is a family of orthogonal polynomials, and when T goes to infinity, $\Pr\{\max_{0\le t\le T}\{Need(t)\} \le k\} \sim Ke^{\alpha T}$, with K a constant and α a root of ω_k with maximal modulus.

Karlin and McGregor [1958] introduce the family of polynomials $\{Q_n(x)\}$ with the properties that $Q_0(x) \equiv 1$ and $-xQ = AQ$, where A is the infinitesimal generator matrix defined so that $A_{k,j}$ is equal to λ_k if $j = k+1$, $-\lambda_k - \mu_k$ if $j = k$, μ_k if $j = k-1$, and 0 otherwise. It turns out that $Q_n(x) = \omega_n(-x)$. This expression gives an extremely simple tool for linking birth-and-death processes to classical families of orthogonal polynomials:

Theorem 2.1. *For the $M/M/1$ process, we have*

$$(\sqrt{a})^n Q_n(\mu x) = T_n(z) - \frac{1}{\sqrt{a}} T_{n-1}(z),$$

where $a = \lambda/\mu$, $z = -(x - a - 1)/\sqrt{a}$ and $\{T_j(a)\}$ is the family of Chebyshev orthogonal polynomials. For the $M/M/\infty$ process, we have

$$Q_j(x) = (-1)^j a^j C_j^{(-a)}(x/\mu),$$

where $\{C_j^{(-a)}(a)\}$ is the family of Poisson Charlier orthogonal polynomials. For several types of linear birth-and-death processes, of the form $\lambda_k = ak + b$, $\mu_k = ck + d$, $Q_j(x)$ can be expressed in terms of either Laguerre polynomials or Meixner polynomials of the second kind.

General birth-and-death processes can also be related to orthogonal polynomials, using the framework of file histories discussed in [Flajolet, Françon, and Vuillemin, 1980].

2.3. Continued Fractions

File histories model the evolution of several classical types of dynamic data structures: stacks (S), priority queues (PQ), linear lists (LL), symbol tables (ST), and dictionaries (D). The data structures are treated as combinatorial objects; their performance characteristics are determined by the *relative* order of the elements they contain, not by the actual values of the elements. Thus, we say that there are $k+1$ ways of inserting a new element into a dictionary of size k, since there are $k+1$ "gaps" where the new element can fit in, relative to the k elements already present. The evolution of the data structure is represented as a path in $\mathbf{Z}^2$ (the x coordinate counts the number of operations, whether they be insertions, deletions or queries, and the y coordinate counts the size), where each step is of the type $(a,b) \to (a+1, b\pm 1)$ (insertion or deletion) or $(a,b) \to (a+1, b)$ (positive or negative query). To each step we associate a certain choice among the possibilities, each equally likely. For example, in priority queue files, deletions can be performed only for the minimum element, so the number of possibilities for a deletion is 1. For purposes of brevity, let us restrict ourselves to the $M/M/\infty$ model in which $\lambda = \mu$. This process is related to histories of symbol tables, in which the number of possibilities for insertion, deletion, and query at level k are equal to $k+1$, 1, and k, respectively [Flajolet, Françon, and Vuillemin, 1980]. We let $H_{j,k}(t)$ be the ordinary generating function of the number of symbol table histories going from level j to k, and we define $H_{j,k}^{\le h}(t)$ similarly except with the histories constrained to have height $\le h$.

Let us consider the bounded process $\lambda_0 = \lambda_1 = \cdots = \lambda_{h-1} = \lambda$, $\lambda_h = 0$, $\mu_k = k\mu$, whose height can never exceed level h (this process can be denoted $M/M/\infty/h$). We define $s_{j,k}^{\le h}(t)$ to be the associated density function for the first passage time to level k. If we call $\sigma_{i,i-1}^{\le h}(s)$ the Laplace transform of $s_{i,i-1}^{\le h}(t)$, then $\sigma_{i,i-1}^{\le h}(s)$ is the solution of the system

$$\sigma_{i,i-1}^{\le h}(s) = \frac{i\mu}{\lambda + i\mu + s - \lambda\sigma_{i+1,i}^{\le h}(s)},$$

for $i < h$, which can be put naturally into the form of a *continued fraction*: $\sigma_{i,i-1}^{\le h}(-\mu(1+1/s))$ equals $i\mu/(\mu+s)$ times

$$\cfrac{1}{1 - \rho s - \cfrac{(i+1)s^2}{1 - (i+1)s - \cfrac{(i+2)s^2}{1 - (i+2)s - \cfrac{\ddots}{1 - (h-1)s - \cfrac{hs^2}{1 - hs}}}}}.$$

All file histories seen so far have their height bounded above or below by some constant. This is due to our concentrating on times of first passage through a state ℓ, which implies that level ℓ must be a barrier for the histories: they must not be allowed to go through state ℓ. But if we now remove the constraint of first passage and consider $P_{k,\ell}(t) = \Pr\{Need(t) = \ell \mid Need(0) = k\}$, in the same way we now get $\pi_{k,\ell}(s)$ the Laplace transform of $P_{k,\ell}(t)$. Taking the inverse Laplace transform will finally yield $s_{j,k}(t)$ and $P_{j,k}(t)$.

2.4. Hashing with Lazy Deletion

The case $H = 1$ in which there is no hashing and a vigilant-deletion strategy is used was analyzed in [Morrison, Shepp, and Van Wyk, 1986]. We can generalize their method to $H > 1$ by considering the appropriate conditional probabilities. Let us for simplicity consider the two bucket case $H = 2$. For bucket i, we define $Use_i(t)$ and $Need_i(t)$ in the obvious way and define $Waste_i(t) = Use_i(t) - Need_i(t)$. We have $Use(t) = Need_1(t) + Need_2(t) + Waste_1(t) + Waste_2(t)$. We can compute the first passage time densities using Laplace transforms and probability techniques, which allows us to calculate the distribution and mean of $\max_{0\le t\le T}\{Use(t)\}$. Discussion will be deferred to the full paper.

2.5. Hermite Polynomials

We consider the non stationary model introduced in [Van Wyk and Vitter, 1986], in which the $2n$ birth times and

deathtimes of the n items are independent uniform random variables from the unit interval. The ith item is born at time $\min\{s_i, t_i\}$ and dies at time $\max\{s_i, t_i\}$. The average queue size $E(Need(t)) = 2nt(1-t)$ attains its maximum $n/2$ at $t = 1/2$. The question of interest is to determine the distribution of the random variable $\max_{0\le t\le 1}\{Need(t)\}$. We shall see that it is the same as the height of a priority queue file history as discussed in [Flajolet, Françon, and Vuillemin, 1981].

By studying involutions with no fixpoints, we can show that our problem is equivalent to determining the distribution of the maximum size of a random priority queue. We denote by $H_{2n}^{\le h}$ the number of priority queue histories of length $2n$ and height $\le h$, and we let $H_{2n}^{\le h}(z)$ be its corresponding generating function. We have $\Pr\{\max_{0\le t\le 1}\{Need(t)\} \le h\} = H_{2n}^{\le h}/(1\cdot 3\cdot\ldots\cdot(2n-1))$. Flajolet [1981] shows that $H_{2n}^{\le h}(z) = Q_{h-1}(z)/Q_h(z)$, where $z^{h+1}Q_h(1/z)$ is the $(h+1)$st orthogonal Hermite polynomial, whose roots are real and distinct. This allows us to get a simple exact expression for $\Pr\{\max_{0\le t\le 1}\{Need(t)\} \le h\}$ and an asymptotic approximation as $n\to\infty$.

3. Optimal Bounds

In this section we prove for the stationary M/G/∞ model that the expected maximum storage needed (that is, the *expected* maximum M/G/∞ queue size) and the expected maximum storage used in excess of that amount are within constant factors, respectively, of the expected storage needed and wasted at any given time. The birth rate is a Poisson process with intensity λ. In the special case of the M/M/∞ model, the lifespans are given by the exponential distribution with mean $1/\mu$. In the general M/G/∞ model, the lifespan distribution is arbitrary, with mean $1/\mu$. The following two theorems are the main results of Section 3:

Theorem 3.1. *We have*

$$E(\max_{0\le t<1}\{Need(t)\}) = O(E(Need)) = O\left(\frac{\lambda}{\mu}\right),$$

under the condition that $\mu = O(\lambda/\log\lambda)$ in the M/M/∞ case, and $\mu = O(\lambda/\log^2\lambda)$ in the general M/G/∞ case.

Theorem 3.2. *Let $\epsilon > 0$ be any constant. Then if the number H of buckets in HwLD is $\Omega((\log\lambda)^{1+\epsilon})$, we have*

$$E(\max_{0\le t\le 1}\{Use(t)\} - \max_{0\le t<1}\{Need(t)\}) = O(E(Use - Need)) = O(H).$$

The restrictions on μ and H in the theorems are extremely weak; they are typically met in geometrical applications, for example [Van Wyk and Vitter, 1986]. In fact, it can be shown that Theorem 3.1 is not true if μ is too large; the restriction is thus partly inherent in the problem. For Theorem 3.2, however, we conjecture that the restriction $H = \Omega((\log\lambda)^{1+\epsilon})$ can be lifted.

We prove Theorem 3.1 in the next section and Theorem 3.2 in Section 3.2. Our approach for both is to approximate the queueing process by a sequence of stages of a discrete analog, which we call *time hashing*. The particular forms of time hashing we use for the two cases are quite different. But they share the common property that the early stages of the time hashing capture most of what is going on in the queueing process; in the later stages the number of slots in the hash table becomes smaller and smaller (and each slot covers a larger span of time) and the contribution becomes less and less.

3.1. Maximum Size of M/G/∞ Queue

This section is devoted to the proof of Theorem 3.1. The number H of buckets in the HwLD implementation does not affect the value of *Need* in any way, so we shall assume in this section that $H = 1$. The distribution of $Need(t)$ is Poisson with mean λ/μ:

Lemma 3.1. *For the M/G/∞ model, we have*

$$\Pr\{Need(t) = i\} = \frac{e^{-\lambda/\mu}\left(\frac{\lambda}{\mu}\right)^i}{i!}.$$

The proof of Theorem 3.1 relies on the following technique we introduce, called *time hashing*: Let K be an integer parameter to be specified later. We shall consider all items that are alive at *some* time during $[0,1]$. Stages $k = 0, 1, 2, \ldots, K$ of time hashing are defined as follows: For $0 \le k \le K$, all items (intervals) that have lifespan in the range $(\frac{1}{\mu}2^{k-1}, \frac{1}{\mu}2^k]$ and that are born in either the unit interval $(0,1]$ or one of the end intervals $(-\frac{1}{\mu}2^k, 0]$ and $(1, \lceil\mu 2^{-k}\rceil\frac{1}{\mu}2^k]$ are put into stage k; in addition, for $k = 0$, the lifespan requirement is weakened so that the lifespan must be in the range $[0, \frac{1}{\mu}]$. Each stage consists of a hash table of $\lceil\mu 2^{-k}\rceil + 1$ slots. The jth slot, for $0 \le j \le \lceil\mu 2^{-k}\rceil$, represents the interval of time $(\frac{1}{\mu}(j-1)2^k, \frac{1}{\mu}j2^k]$. An item in stage k is placed into the slot corresponding to its birthtime. We also define a special stage $K+1$ as follows: Slot 0 consists of all items born in $[0,1]$ with lifespan $> \frac{1}{\mu}2^{k+1}$; the remaining $\lceil\mu 2^{-(k+1)}\rceil$ slots are left empty.

We define $N_k(j)$ to be the number of stage k items in slot j. The following fundamental relation bounds $\max_{0\le t\le 1}\{Need(t)\}$ by the sum of the expected maximum slot occupancies in time hashing:

Lemma 3.2. *We have*

$$\max_{0\le t\le 1}\{Need(t)\} \le 2\sum_{0\le k\le K+1}\max_{0\le j\le\lceil\mu 2^{-k}\rceil}\{N_k(j)\}$$

The M/M/∞ Case. First we shall handle the M/M/∞ case, in which the lifetimes are exponentially distributed with mean $1/\mu$. The restriction on μ in Theorem 3.1 is slightly weaker in this case than in the general M/G/∞ case. In this subsection we assume that we are dealing with the M/M/∞ model and that $\mu = O(\lambda/\log\lambda)$. We define the stage parameter K to be $\lceil\lg\ln\mu^{-1}\rceil$.

Lemma 3.3. *The expected number of items in stage $K+1$ is*

$$E(N_{K+1}(0)) = E(\max_{0\le j\le\lceil\mu 2^{-K}\rceil}\{N_{K+1}(j)\}) \le \frac{\lambda}{\mu}$$

Lemma 3.4. *For $0 \le k \le K$, let n be the average number of items in stage k, and let $m = \lceil\mu 2^{-k}\rceil + 1$. In*

the number of slots in the time hashing table of stage k. Then the number $N_k(j)$ of items in slot j of stage k is Poisson distributed with mean $\alpha = n/m$, where

$$\alpha = \begin{cases} \frac{\lambda}{\mu} 2^k (e^{-2^{k-1}} - e^{-2^k}), & \text{if } 1 \le k \le K; \\ \frac{\lambda}{\mu}(1 - e^{-1}), & \text{if } k = 0. \end{cases}$$

Lemma 3.5. *The expected maximum occupancy of the slots in stage k, $0 \le k \le K$, is*

$$E(\max_{0 \le j \le \lceil \mu 2^{-k} \rceil} \{N_k(j)\}) = O\left(\frac{\lambda}{\mu 2^k}\right).$$

The proof of Lemma 3.5 makes use of the following lemma and corollary. They give us an upper bound in an easy way for the expected maximum slot occupancy in hashing. The lemma is phrased for general slot occupancies X_j that are not assumed to be independent; when the occupancies are independent or satisfy a certain property, the bound in the corollary is obtained.

Lemma 3.6. *For random variables $X_1, \ldots, X_m$, if $\Pr\{X_j > b\} \le 1/(nm)$, for all $1 \le j \le m$, where $n = E(\sum_j X_j)$, then we have*

$$E(\max_{1 \le j \le m} \{X_j\}) \le b + \frac{1}{n} E(\max_{1 \le j \le m} \{X_j\} \mid \max_{1 \le j \le m} \{X_j\} > b).$$

Corollary 3.1. *If in addition to the assumption required for Lemma 3.6 we also have*

$$E(\max_{1 \le j \le m} \{X_j\} \mid \max_{1 \le j \le m} \{X_j\} > b) \le E(\max_{1 \le j \le m} \{X_j\} \mid X_1 > b),$$

then

$$E(\max_{1 \le j \le m} \{X_j\}) \le b + \frac{1}{n} E(\max_{1 \le j \le m} \{X_j\} \mid X_1 > b).$$

The rest of the proof of Theorem 3.1 for the M/M/∞ case consists of taking expectations in the expression of Lemma 3.2 and substituting the bounds from Lemmas 3.3 and 3.5, which gives a convergent geometric series.

The M/G/∞ Case. In this subsection we assume that $\mu = O(\lambda/\log^2 \lambda)$. For the case of the M/G/∞ model, the distribution of lifetimes is allowed to be an arbitrary one with mean $1/\mu$. So in particular the approach we used above for M/M/∞ (namely, Lemma 3.5) will not work; for each given value of k, stage k could contribute as much as $\Omega(\lambda/\mu)$ to $E(\max_{0 \le j \le \lceil \mu 2^{-k} \rceil} \{N_k(j)\})$. Instead we use the following important correspondence between the average slot occupancies and $E(\textit{Need})$:

Lemma 3.7. *Let $\alpha_k = E(N_k(0))$ be the average number of items in slot 0 of stage k. Then*

$$\frac{1}{2} \sum_{1 \le k < K+1} \alpha_k \le \frac{\lambda}{\mu}.$$

We use time hashing as before, but with the stage parameter set to $K = \lceil \lg \mu \rceil$. There are $\lceil \mu 2^{-k} \rceil + 1 \le \mu + 2$ slots in stage k, for each $0 \le k < K$. An easy application of Corollary 3.1 gives us the following key lemma, which is the basis for the proof of Theorem 3.1 for the M/M/∞ case.

3.2. Optimal Bounds on Waste in HwLD

To prove Theorem 3.2, we derive an upper bound for $E(\max_t\{\textit{Waste}(t)\})$, where $\textit{Waste}(t) = \textit{Use}(t) - \textit{Need}(t)$ is the number of dead items that are still in the HwLD data structure at time t. This therefore gives an upper bound on $E(\max_t\{\textit{Use}(t)\} - \max_t\{\textit{Need}(t)\})$. It is important to note that the former quantity is usually larger than the latter, because $\textit{Use}(t)$ and $\textit{Need}(t)$ typically do not attain their maxima at the same time t.

To bound the expected maximum waste, we use a time hashing of a different nature than in Section 3.1. The stages are numbered $k = 0, 1, \ldots, K+1$, and each of the H buckets has its own set of stages. The hash table for each bucket for stage k has $\lceil \frac{\lambda}{H} 2^{-(k+1)} \rceil$ slots. The jth slot, for $0 \le j \le \lceil \frac{\lambda}{H} 2^{-(k+1)} \rceil - 1$, represents the time interval $[j 2^{k+1} \frac{H}{\lambda}, (j+1) 2^{k+1} \frac{H}{\lambda}]$. The first half of each slot is called the *death zone*, and the second half is called the *twilight zone*. For each stage, one entry is put into its jth slot for every death in the death zone of its jth slot, with the extra requirement that there are no births in the twilight zone of the jth slot; if there is a birth in the twilight zone, no entries are placed into the jth slot.

In addition, stages 0 and $K+1$ are supplemented as follows: In stage 0, an entry is put into the jth slot for every death in the death zone, regardless of whether there have been no births in the twilight zone. In stage $K+1$, we move all the entries into slot 0 from the other slots.

We let $w_{h,k}(j)$ denote the slot occupancy for the jth slot in the time hashing table for bucket h in the kth stage. We define $W_k(j)$ to be the total number of entries in the jth slots of the hash tables for buckets $1, 2, \ldots, H$:

$$W_k(j) = \sum_{1 \le h \le H} w_{h,k}(j).$$

We set the stage parameter K to be $K = \lceil \lg \ln(\lambda/H) \rceil$.

For completeness, we should mention that there is a total of four instances of time hashing, not just the one defined above. The second instance of time hashing is defined in an identical way, except that the time intervals of the slots are offset $\frac{H}{\lambda} 2^k$ from the time intervals of the instance defined above. In addition to these two instances, we consider two "reverse" instances, in which time is viewed backwards: we start at time $t = 1$ and end at time $t = 0$, and we process each death as a birth and vice-versa. Without loss of generality we shall discuss only the first instance of time hashing, as defined in the previous paragraphs, and introduce an extra factor of 4 into our bounds, where appropriate.

A key observation for the derivation is that the death rate in the M/G/∞ model is a Poisson process with the same intensity as the birth rate. This follows because the M/G/∞ model is symmetric and stationary, and thus also reversible [Kelly, 1979]. The following lemma is the basis for our proof of Theorem 3.2:

Lemma 3.9. *We have*

$$\max_{0 < t < 1} \{\textit{Waste}(t)\} < 4 \sum_{0 < k < K+1} \max_{0 < j < \lceil \frac{\lambda}{H} 2^{-(k+1)} \rceil} W_k(j)\}$$

We shall prove Theorem 3.2 by bounding the sum in Lemma 3.9 by $O(E(Waste)) = O(H)$. A big difference between this application of time hashing and the ones we used in Section 3.1 is that the random variables $w_{h,k}(j)$ (and hence also $W_k(j)$) are almost always 0 as k grows. We have $\Pr\{w_{h,k}(j) = 0\} \approx 1 - e^{-2^k}$. This causes the maximum slot occupancy to behave wildly. In fact, to get our bound, it is not enough to bound $E(\max_j\{w_{h,k}(j)\})$ and then multiply by H, because the result will be too large: the load factor in the analysis of $\max_j\{w_{h,k}(j)\}$ is too small, and the ratio between the average maximum slot occupancy and the average slot occupancy is no longer $O(1)$. The solution is to consider the H buckets *in toto* and to bound $E(\max_j\{W_k(j)\})$ directly. We do that by computing the moment generating function of $\max_{0\le j\le\lceil\frac{\lambda}{H}2^{-k+1}\rceil}\{W_k(j)\}$ and then applying Corollary 3.1 using Chernoff's bound.

Lemma 3.10. *The expected number of entries in stage $K+1$ is*

$$E(W_{K+1}(0)) = E\Big(\max_{0\le j\le\lceil\frac{\lambda}{H}2^{-K+1}\rceil}\{W_{K+1}(j)\}\Big) = O(H).$$

Lemma 3.11. *The expected maximum occupancy of the slots in stage k, $0 \le k \le K$, is*

$$E\Big(\max_{0\le j\le\lceil\frac{\lambda}{H}2^{-k}\rceil}\{W_k(j)\}\Big) = O\left(\frac{H}{2^{k/d}}\right),$$

if $H = \Omega((\log\lambda)^{1+1/d})$.

Theorem 3.2 follows by combining Lemmas 3.9, 3.11, and 3.12, and summing on k.

4. Conclusions

The maximum size attained by a queue over time is a basic notion in stochastic processes and queueing theory. In terms of data structures, if we model the insertions and deletions of elements as the birth and death of items in a queue, then the maximum queue size is the maximum size of the data structure. Our conclusions come in two forms: First, we have used in a natural way a variety of algebraic and analytical techniques to obtain exact formulas for the distribution of the maximum size of queues for birth-and-death processes and for hashing with lazy deletion (HwLD). Our solutions are amenable to numerical calculation and some asymptotics. The formulas for several different models are related in that the relevant transform in each case can be expressed as a ratio of classical orthogonal polynomials.

Second, we have answered some open questions in *queueing theory* using discrete, non-queueing theory techniques. We have obtained optimal big-oh bounds on the expected maximum queue size for the M/G/∞ process and for HwLD. We prove for HwLD that the expected maximum amount of needed space (that is, the maximum size of the M/G/∞ queue, on the average) and the expected maximum amount of space used by HwLD above the optimal amount are within small constant factors, respectively, of the average space needed and wasted at any given time. Our techniques also appear to be applicable to the M/M/1 model, which introduces several interesting new facets to the problem.

Current work is aimed at removing the the condition $H = \Omega(\log\lambda)$ from Theorem 3.2. The proof technique, though, has to be different, because it is easy to show for $H = 1$ that $\max_{0\le t\le 1}\{Waste(t)\}$ has unbounded expectation. Another problem being worked on is to determine the constant factors inherent in the big-oh bounds. Preliminary results suggest that the constants in Theorems 3.1 and 3.2 are asymptotically 1 under general conditions.

Acknowledgements. The authors would like to thank François Baccelli, Guy Fayolle, Philippe Flajolet, and Claude Puech for interesting discussions. Support for the first author was provided in part by a Procter Fellowship. Support for the second author was provided in part by NSF research grant DCR 84-03613, by an NSF Presidential Young Investigator Award with matching funds from an IBM Faculty Development Award and an AT&T research grant, and by a Guggenheim Fellowship.

References

P. Flajolet, "Analyse d'algorithmes de manipulation d'arbres et de fichiers," *Cahiers du Bureau Universitaire de Recherche Opérationnelle*, **34–35** (1981), 1–209.

P. Flajolet, J. François, and J. Vuillemin, "Sequence of Operations Analysis for Dynamic Data Structures," *Journal of Algorithms*, **1**(2) (June 1980), 111–141.

S. Karlin and J. M. McGregor, "Linear Growth Birth and Death Processes," *Journal of Mathematics and Mechanics*, **7**, 4 (1958).

F. P. Kelly, *Reversibility and Stochastic Networks*, Series in Probability and Mathematical Statistics, Wiley & Sons, Chichester (1979).

L. Kleinrock, *Queueing Systems*, Volume I: *Theory*, Wiley & Sons, New York (1975).

J. Morrison, L. A. Shepp, and C. J. Van Wyk, "A Queueing Analysis of Hashing with Lazy Deletion," *SIAM Journal on Computing* **16**, 6 (December 1987), 1155–1164.

T. Ottmann and D. Wood, "Space Economical Plane-Sweep Algorithms," *Computer Vision, Graphics, and Image Processing*, **34** (1986), 35–51.

T. G. Szymanski and C. J. Van Wyk, "Space Efficient Algorithms for VLSI Artwork Analysis," *Proceedings of the 20th IEEE Design Automation Conference* (June 1983), 743–749.

C. J. Van Wyk and J. S. Vitter, "The Complexity of Hashing with Lazy Deletion," *Algorithmica*, **1**(1) (March 1986), 17–29.

CLASSIFYING LINEAR MIXTURES, WITH AN APPLICATION TO HIGH RESOLUTION GAS CHROMATOGRAPHY

William S. Rayens, University of Kentucky

1. INTRODUCTION

1.1 Overview

Consider g groups, each of which can be characterized in terms of p particular variables. Suppose a test observation y is a "linear mixture" in the sense that each of the p variables associated with y can be characterized as a convex combination of the corresponding variables in these component groups. The weights defining this convex combination will be called "mixing proportions". The test observation is "classified" when the mixture constituents are identified and the mixing proportions are estimated.

In this paper we propose a model which seeks to classify linear mixtures. Section 2 contains the motivation for, and an outline of the model development. Section 3 contains the details and results of the application of the model to the problem of identifying the constituents in polychlorinated biphenyl samples. Finally, section 4 contains a statement of our conclusions, and section 5 briefly mentions the computer routine used to implement the methodology.

1.2 Application Context

Polychlorinated biphenyls (PCBs) occurring in the environment of the United States originate from one or more of nine industrial products: *Aroclors* (registered trademark of the Monsanto Corporation). Each of these nine can be characterized by a particular set of constituents and their relative concentrations which are determined by gas chromatography.[2] These constituents differ by the location of chlorine atoms along the carbon chain associated with a biphenyl molecule. Theoretically, there are 209 distinguishable arrangements; far fewer are generally available in practice. Further, an environmental or biological specimen can be characterized as a weighted average of the constituent concentrations present in the component Aroclors, in which the weights are the mixing proportions. That is, the chromatogram associated with a mixture is essentially a weighted average of the chromatograms associated with the component Aroclors present in the mixture.

We had access to a training set, Y, consisting of six runs for each of the nine pure Aroclors. Unique, identifiable peaks in the chromatograms led to the development of 93 concentration variates that correspond to relative concentrations of individual PCBs or conjoint PCBs that coelute.[3] Hence, Y contains $N=54$ observations (rows) and $p=93$ variables (columns). Rows 1-6 of Y correspond to the runs on Aroclor 1; rows 7-12 are the runs on Aroclor 2, etc. The aforementioned reference furnishes the details concerning the chemical and detection methods that ultimately led to the training set Y.

2. DEVELOPMENT OF THE MODEL

2.1 Assumptions

Suppose a random vector $y \in \mathbb{R}^p$ is p-variable normal, $N_p(\mu(\gamma),\Sigma)$, where $\mu(\gamma)=\sum_{i=1}^{g}\gamma_i\mu_i$; $\gamma^t=(\gamma_1,\ldots,\gamma_g)$ is the vector of mixing proportions, so $\sum_{i=1}^{g}\gamma_i=1$, and $\gamma_i \geq 0$ for all i; Σ is assumed to be positive definite, and $\mu_i \in \mathbb{R}^p$ for $i=1,\ldots,g$. Our objective is to estimate γ. The model assumptions can be subjected to criticism but in the final analysis the assessment of the methodology will be made not on the validity of the model assumptions, but on how well the procedures work on real data.

In the PCB context γ_j interprets as the concentration of the j^{th} Aroclor in the mixture. Likewise the vector μ_j represents the pure chromatogram corresponding to the j^{th} Aroclor and Σ is the covariance matrix of the chromatograms. In developing our model we will first consider the covariance matrix and the pure chromatograms to be known.

2.2 Classification with Σ and $\mu_1,\ldots,\mu_g$ known.

When Σ and $\mu_1,\ldots,\mu_g$ are known, we can use maximum likelihood to estimate γ. The likelihood function associated with $\mu(\gamma)$ is:

$$\Lambda(\mu(\gamma)) = 1/\left((2\pi)^{p/2}\det(\Sigma)\right) \exp\left[-(1/2)(y-\mu(\gamma))^t\Sigma^{-1}(y-\mu(\gamma))\right].$$

As a function of γ the maximum of this expression is achieved where $(y-\mu(\gamma))^t\Sigma^{-1}(y-\mu(\gamma))$ (the Mahalanobis distance from y to $\mu(\gamma)$) is minimized. Define

$$Q \equiv \left\{\gamma \in \mathbb{R}^g : \gamma^t=(\gamma_1,\ldots,\gamma_g),\ \sum_{i=1}^{g}\gamma_i=1,\ \gamma_i \geq 0\ \forall i\right\}$$

The restrictions $\mu(\gamma)=\sum_{i=1}^{g}\gamma_i\mu_i$ and $\gamma \in Q$ constrain $\mu(\gamma)$ to a simplex having $\mu_1,\ldots,\mu_g$ as vertices. Hence, a maximum likelihood estimate of γ, say $\hat{\gamma}$, is found by locating the point on this simplex, $\mu(\hat{\gamma})$, that is closest to y in the sense of Mahalanobis distance in $\mathbb{R}^p$.

Suppose $L \equiv \text{span}\left\{\Sigma^{-(1/2)}(\mu_i-\mu_1)\right\}_{i=2}^{g}$ and $B \equiv$ any matrix matrix whose columns form an orthonormal basis for L. The following theorem shows that the transformation $B^t\Sigma^{-(1/2)}$: $\mathbb{R}^p \rightarrow \mathbb{R}^{g-1}$ reduces the problems of finding

$\hat{\gamma}$ in terms of Mahalanobis distance in $\mathbf{R}^p$ to an identical problem involving Euclidean distance in $\mathbf{R}^{g-1}$.

Notation:

$SP = (g-1)$–simplex having vertex set

$$\left\{B^t\Sigma^{-(1/2)}(\mu_i)\right\}_{i=1}^{g} \equiv \{v_i\}$$

$$z \equiv B^t\Sigma^{-(1/2)}y$$

$$\hat{z} \equiv \text{point on } SP \text{ closest to } z$$

$$\beta^t = (\beta_1, \ldots, \beta_g) \in Q, \text{ the barycentric coordinates of } \hat{z} \text{ relative to } SP.$$

Note: $B(B^tB)^{-1}B^t\Sigma^{-(1/2)}(\mu_i) = BB^t\Sigma^{-(1/2)}(\mu_i) = B(v_i)$. Hence, the $(g-1)$ components of v_i represent the coordinates (with respect to the basis B) of the orthogonal projection of $\Sigma^{-(1/2)}(\mu_i)$ onto L.

Theorem 1:

β is a maximum likelihood estimate of γ, subject to $\gamma \in Q$. That is, $\beta = \hat{\gamma}$.

Proof: We need to introduce some more notation and make two observations.

Additional Notation:

$$y_{est} \equiv \sum_{i=1}^{g}\gamma_i\mu_i = \mu_1 + \sum_{i=2}^{g}\gamma_i(\mu_i - \mu_1)$$

$$w_1 \equiv \Sigma^{-(1/2)}\mu_1$$

$$z_1 \equiv B^t w_1$$

$$w^* \equiv \Sigma^{-(1/2)}y_{est} - w_1$$

$$z_{est} \equiv B^t\Sigma^{-(1/2)}y_{est} = B^t w^* + z_1$$

$$w \equiv \Sigma^{-(1/2)}y - w_1$$

$\|v\| \equiv$ length of the vector v, in terms of the usual Euclidean inner product.

Note 1:
$$\begin{aligned} w^* &\equiv \Sigma^{-(1/2)}y_{est} - w_1 \\ &= \Sigma^{-(1/2)}\left[\mu_1 + \sum_{i=2}^{g}\gamma_i(\mu_i - \mu_1)\right] - w_1 \\ &= \sum_{i=2}^{g}\gamma_i\left[\Sigma^{-(1/2)}(\mu_i - \mu_1)\right] \in L. \end{aligned}$$

It follows that $B(B^tB)^{-1}B^t(w^*) = BB^t(w^*) =$ "orthogonal projection of w^* onto L" $= w^*$.

Note 2: if $w^* \in L$ and $w \in \mathbf{R}^p$, then
$\|w - w^*\|^2 = \|w - BB^t w\|^2 + \|BB^t w - w^*\|^2$.

A maximum likelihood estimator is calculated as follows:

$$\begin{aligned} &\min_{\gamma\in Q}\left[(y - y_{est})^t\Sigma^{-1}(y - y_{est})\right] \\ &= \min_{\gamma\in Q}\left[y - y_{est})^t\Sigma^{-(1/2)}\Sigma^{-(1/2)}(y - y_{est})\right] \\ &= \min_{\gamma\in Q}\left[(\Sigma^{-(1/2)}y - \Sigma^{-(1/2)}y_{est})^t(\Sigma^{-(1/2)}y - \Sigma^{-(1/2)}y_{est})\right] \\ &= \min_{\gamma\in Q}\left[(w + w_1) - (w^* + w_1)\right]^t\left[(w + w_1) - (w^* + w_1)\right] \\ &= \min_{\gamma\in Q}\left[(w - w^*)^t(w - w^*)\right] \\ &= \min_{\gamma\in Q}\|w - w^*\|^2 \\ &= \min_{\gamma\in Q}\left[\|w - BB^t w\|^2 + \|BB^t w - w^*\|^2\right] \text{ (Note 2)}. \end{aligned}$$

Since $\|w - BB^t w\|^2$ is independent of γ, the calculation of a maximum likelihood estimator reduces to finding:

$$\begin{aligned} &\min_{\gamma\in Q}\|BB^t w - w^*\|^2 \\ &= \min_{\gamma\in Q}\|BB^t w - BB^t w^*\|^2 \text{ (Note 1)} \\ &= \min_{\gamma\in Q}\|B(z - z_1) - B(z_{est} - z_1)\|^2 \\ &= \min_{\gamma\in Q}\|Bz - Bz_{est}\|^2 \\ &= \min_{\gamma\in Q}(Bz - Bz_{est})^t(Bz - Bz_{est}) \\ &= \min_{\gamma\in Q}(z - z_{est})^t B^t B(z - z_{est}) \\ &= \min_{\gamma\in Q}(z - z_{est})^t(z - z_{est}) \\ &= \min_{\gamma\in Q}\|z - z_{est}\|^2. \end{aligned}$$

Hence, z_{est} must be chosen to be the point on SP that is closest to z. That is, $z_{est} = \hat{z}$, and $\hat{\gamma} = \beta$ as claimed. □

This result makes it clear how γ can be estimated, and, hence, what our model will be (when all the parameters μ_i and Σ are known). First, we form the simplex SP.

Next, given a $p\times 1$ vector y as the test observation, we calculate the $(g-1)\times 1$ "score" $z=B^t\Sigma^{-(1/2)}y$ and identify z with the closest point on SP, say $\hat{z}$. Finally we calculate the barycentric coordinates of $\hat{z}$ with respect to SP and use these as estimates of the unknown mixing proportions.

2.4 Classification with $\mu_1, \ldots, \mu_g$ and Σ unknown

In practical situations such as the PCB application the group population means (pure chromatograms), as well as Σ (the covariance matrix), will usually be unknown. *However, we can still reach conclusions similar* to those in the previous section. That is, suppose $Y_{N\times p}$ denotes the training set mentioned above, representing N total observations, n_i from the i^{th} group, $\sum_{i=1}^{g} n_i = N$. The existence of this training set permits the estimation of Σ and all the μ_i. For instance, Suppose μ_i is estimated by $\bar{y}_i$, the sample mean of the i^{th} group; and Σ is estimated by $S\equiv\left[1/(n-g)\right]E$, where E is the within-groups sums-of-squares and cross-products matrix associated with Y. If we view the likelihood function Λ as a function of γ alone, we can replace Σ by S and μ_i by $\bar{y}_i$ in the above notation and immediately reach the *conclusion in* Theorem 1.

There is an important question left to be answered. In the practical setting, where does one find a B matrix satisfying the above requirements? By answering this question, we will establish a strong connection between the above ideas and linear discriminant analysis. Consider the following notation:

$\bar{y} \equiv$ grand mean $=(1/N)\sum_{i=1}^{g} n_i \bar{y}_i$

$L \equiv \text{span}\left\{E^{-(1/2)}(\bar{y}_i-\bar{y}_1)\right\}$

$H \equiv \sum_{i=1}^{g} n_i(\bar{y}_i-\bar{y})(\bar{y}_i-\bar{y})^t$

$B \equiv (b_1 \cdots b_{g-1})_{p\times(g-1)}$, matrix whose columns are the $g-1$ eigenvectors corresponding to the $(g-1)$-nonzero eigenvalues of $E^{-(1/2)}H\,E^{-(1/2)}$

$B^* \equiv \text{span}\{b_i\}_1^{g-1}$

$M \equiv$ matrix whose columns are the $g-1$ eigenvectors corresponding to the $(g-1)$-nonzero eigenvalues of $E^{-1}H$. That is, M is the matrix which results from a standard linear discriminant analysis.

$Z_{N\times(g-1)} \equiv$ matrix of discriminant scores $= YM$.

The following theorems, proved in reference 4, establish the connection.

Theorem 2:

$$M^t = B^t E^{-(1/2)}, \text{ and } B^t B = I_{(g-1)\times(g-1)}.$$

Theorem 3:

$$B^* = L\,.$$

These results point out that the columns of B are an orthonormal basis for L; also, the transformation resulting from a linear discriminant analysis has the form $M^t = B^t E^{-(1/2)}$. Hence, we arrive at the following procedure for estimating the unknown mixing proportions:

Step 1 — form the simplex SP, defined by the vertex set $\{\bar{z}_i\}_{i=1}^{g}$, where $\bar{z}_i$ is the sample mean of the i^{th} group of discriminant scores.

Step 2 — given y is a test observation (mixture), admitting discriminant score $z=M^t y$, find the point, say $\hat{z}$, on SP which is closest to z.

Step 3 — use the barycentric coordinates of $\hat{z}$, given by $\beta^t=(\beta_1, \ldots, \beta_g)$, to estimate the unknown mixing proportions.

2.4 Location of the closest point

We have shown that the vector of mixing proportions can be rigorously estimated provided we can find the point on a given simplex in $\mathbb{R}^{(g-1)}$ that is closest to a fixed point $z\in\mathbb{R}^{(g-1)}$. We can adapt a fairly common nonlinear programming scheme (a "gradient projection" technique) to solve this problem. So that our general direction is not lost, we direct the reader to reference 4 for details. The fact is, the desired closest point can be found in a (relatively easy) iterative fashion.

3. RESULTS OF PCB APPLICATION

The training set $Y_{54\cdot 93}$ is not appropriate for a discriminant analysis as it stands, because the column dimension is too large. We therefore used a principal component analysis as a preliminary step to reduce the column dimension. Then, we performed a linear discriminant analysis to obtain a matrix of discriminant scores $Z_{54\cdot 8}$. From this we formed the 9 vertex vectors of the simplex SP in $\mathbb{R}^8$ by calculating group means.

For use in testing the effectiveness of our model, we had access to a matrix consisting of several runs on the same three-component mixture. Using methods of

gravimetric measuring, pure samples of Aroclors 1, 6 and 7 were weighed in the relative proportions of 2.5:2:1 (respectively), and then mixed. That is, the mixture theoretically consisted of 45.5% Aroclor 1, 36.3% Aroclor 6, and 18.2% Aroclor 7. Using methods of high resolution gas chromatography, 38 runs were made on this mixture, and the same 93 variates as in the training set were isolated.

These pseudo-unknowns were treated as test observations and the 38 corresponding discriminant scores were obtained. These 38 points in $\mathbb{R}^8$ were then identified with the corresponding 38 closest points on SP. The barycentric coordinates of these latter points were calculated with respect to SP and used as estimates of the mixing proportions. The classification results from our model are shown in Table 1.

The following comments highlight the important features of these results:[2]

- The Aroclors actually present in the mixture (numbers 1, 6, and 7) were identified correctly (i.e., had a positive barycentric coordinate) in every case except one (run number 5).
- Exactly these three Aroclors were identified in 12 of the 38 runs. Of the remaining 26 cases, the estimated contributions from Aroclors other than 1, 6, or 7 totaled less than 1 percent in 9 cases. (Except for run number 14, these small contributions were always associated with a single Aroclor—namely, Aroclor 8.) Thus, in 32 of the 38 cases, the contributions of Aroclors 1, 6, and 7 are estimated to exceed 95 percent of the complete mixture.
- Examination of the pattern of the ESS measure reveals that the first 31 cases (i.e., rows in the table) are all relatively similar in terms of the accuracy achieved; the last 7 cases exhibit substantially higher values of the ESS. This suggests that the concentration data for these 7 runs differs in some significant respect from the other runs. The variability of the classification results among these 7 runs (last seven rows) suggests that several different types of anamolies may be present in these runs, as opposed to a single type. Five of these cases correspond to runs made late in the study, as identified by the run number. In fact, five of the last six tripartite runs fall into this group of seven (runs 33, 34, 36, 37, 38). This suggests that some type(s) of instrument degradation and/or contamination may be responsible for the poorer performance on these runs.
- Among the first 31 cases shown in the table, the estimated percentages are quite consistent, as summarized in Table 2.
- Despite the consistency of the estimates evidenced in Table 2, it is clear that there is a discrepancy of about eight percentage points between the estimates of Aroclor 1 and the gravimetric weight. The Aroclor 7 estimates admit a similar discrepancy. The source of the bias is not clear, but the consistency of the estimates is encouraging. It suggests that accurate results may be obtainable by adjusting for the bias. Further study is needed to resolve this matter.

4. CONCLUSIONS

The results presented in section 3 clearly support the potential worth of our model. Classical discriminant techniques are principally concerned with the classification of a test observation into one group. Many recent methods—eg. SIMCA[6] and "classification trees"[1]—are also directed to this purpose. In these methods statements are available concerning the probability of membership in a certain class. However, it is unclear how to translate this uncertainty into a statement concerning mixing proportions. The model we have developed is more focused and presents easily interpreted results.

Well-known constrained least-squares techniques are certainly applicable to the problem we have addressed. In fact, we are essentially performing a least-squares analysis once the discriminant scores are obtained. However, the use of discriminant analysis to initially "best separate" the groups is found to be a useful and intuitive step.

5. COMPUTER ROUTINE

Our model employed computer routines programmed in SAS and executed under SAS Release 82.4 at Triangle Universities Computation Center (TUCC), Research Triangle Park, N.C. The routines are extremely flexible and essentially allow the entire procedure outlined in section 2 to be automatically performed, including the discriminant analysis and a principal component analysis, if necessary or desired. Further, several refinements and methodological extensions not mentioned in this article are available as options.[4]

ACKNOWLEDGMENTS

This work was supported in part by a grant from the

United States Environmental Protection Agency, Statistical Policy Branch, Chemicals and Statistical Policy Division, Office of Standards and Regulations, 401 M Street, SW, Washington, D.C., 20460—contract numbers 68-01-6826 and 68-01-5915. We are indebted to C.A. Clayton, and E.D. Pellizzari, of the Research Triangle Institute, Research Triangle Park, N.C., 27709, as well as D.S. Burdick of Duke University for their assistance.

DISCLAIMER

REFERENCES

1. Breiman L; Friedman JH; Olshen RA; Stone CJ. *Classification and Regression Trees.* Belmont, California. Wordsworth Statistics/Probability Series (1984).
2. Clayton CA; Pellizzari ED; Rayens WS; Burdick DS. "Development and Demonstration of a Pattern Recognition Technique for identifying Constituents of Polychlorinated Biphenyl Mixtures," *RTI Draft Report—EPA Contract 68-01-6826* (1985).
3. Pellizzari ED; Moseley MA. "Evaluation of SIMCA Pattern Recognition Using Polychlorinated Biphenyl Data Sets," *RTI Final Report—EPA Contract 68-03-3009* (1984).
4. Rayens WS. "A Model for Classifying Linear Mixtures," Ph.D. Dissertation. Duke University (1986).
5. *SAS User's Guide: Statistics (Version 5 Edition).* Cary, North Carolina: SAS Institute, Inc. (1985).
6. Wold S; Albano C; Dunn WJ; Edlund U; Esbensen K; Geladi P; Hellberg S; Johansson E; Lindberg WJ; Sjostrom M. "Multivariate Data Analysis in Chemistry", in *Chemometrics: Mathematics and Statistics in Chemistry.*
   BR Kowalski, ed. Hingham, Massachusetts: Kluwer Academic Publishers (1984).

TABLE 1 — Classification Results for 38 Runs on the Three Component Mixture

	Aroclor Number									
Run	1	2	3	4	5	6	7	8	9	ESS
4	50.7	· · ·	· · ·	· · ·	· · ·	37.0	10.0	2.2	· · ·	36.5
27	52.3	· · ·	· · ·	· · ·	· · ·	37.6	10.2	· · ·	·	112.1
15	52.8	· · ·	· · ·	· · ·	· · ·	36.5	10.3	0.4	·	116.3
28	52.7	· · ·	· · ·	· · ·	· · ·	37.2	10.1	· · ·	·	118.5
26	52.6	· · ·	· · ·	· · ·	· · ·	37.0	10.0	0.4	· · ·	118.6
35	51.7	· · ·	· · ·	· · ·	· · ·	36.8	9.4	2.1		120.7
25	53.4	· · ·	· · ·	· · ·	·	36.0	10.6			120.7
10	52.0	· · ·	· · ·	· · ·	· · ·	36.7	9.4	· · ·		123.7
9	53.5	· · ·	· · ·	· · ·	· · ·	36.4	10.1	· · ·	· · ·	130.0
11	52.7	· · ·	· · ·	· · ·	· · ·	37.9	9.4	· · ·	·	132.0
23	53.8	· · ·	· · ·	· · ·	· · ·	36.0	10.2	· · ·	· · ·	133.5
22	53.8	· · ·	· · ·	· · ·	· · ·	35.9	10.2	0.1	·	133.6
7	50.2	· · ·	· · ·	· · ·	· · ·	38.0	8.3	3.5	· · ·	135.1
24	53.6	· · ·	· · ·	· · ·	· · ·	36.5	9.7	0.1	· · ·	138.8
3	53.3	· · ·	· · ·	· · ·	· · ·	34.5	9.9	2.3	· · ·	138.9
6	52.3	· · ·	· · ·	· · ·	· · ·	36.7	8.8	2.1	· · ·	139.4
19	53.8	· · ·	· · ·	· · ·	· · ·	35.8	9.8	0.6	· · ·	140.6
1	54.6	· · ·	· · ·	· · ·	· · ·	34.6	10.8	· · ·	· · ·	141.2
21	54.8	· · ·	· · ·	· · ·	· · ·	34.3	10.4	0.5	· · ·	152.4
13	55.1	· · ·	· · ·	· · ·	· · ·	33.8	10.8	0.3	· · ·	154.2
20	54.9	· · ·	· · ·	· · ·	· · ·	35.0	10.2	· · ·	· · ·	154.8
2	51.6	· · ·	· · ·	· · ·	· · ·	35.4	8.2	4.7	· · ·	160.4
18	54.7	· · ·	· · ·	· · ·	· · ·	34.7	9.5	1.1	·	164.8
16	55.5	· · ·	· · ·	· · ·	· · ·	33.8	10.5	0.2	· · ·	166.5
8	53.7	· · ·	· · ·	· · ·	· · ·	38.3	8.0	· · ·	·	175.4
31	56.3	· · ·	· · ·	· · ·	·	31.5	12.2	· · ·	· · ·	177.1
32	56.4	· · ·	· · ·	· · ·	· · ·	31.9	11.6	· · ·		183.0
29	54.4	· · ·	· · ·	· · ·	· · ·	34.8	8.2	2.6	·	188.9
30	56.8	· · ·	· · ·	· · ·	· · ·	31.0	12.2	·		193.3
17	55.2	· · ·	· · ·	· · ·	· · ·	34.6	8.5	1.7		194.7
12	56.3	·	· · ·	· · ·	· · ·	33.9	9.3	0.5	·	202.8
14	60.6	· · ·	·	· · ·	1.4	29.2	7.9	0.2	0.7	288.9
34	20.5	3.7	· · ·	0.2	16.8	34.2	10.1	4.5		609.9
36	25.9	4.8		· · ·	19.1	34.7	14.9	·	0.3	795.4
33	25.1	4.3		8.2	11.8	32.9	10	4.1	0.2	806.9
5	45.6	·	·		· · ·	23.8	· · ·	24.5	6.1	1125.9
37	14.9	5.7	· · ·	7.9	3.5	50.3	12.0	·	5.7	1305.7
38	12.0	4.8	·	7.9	1.3	56.4	10.3	·	7.3	1723.2

TABLE 2 — Summary of Consistency Among the First 31 Runs Listed in Table 1

	Aroclor			
	1	6	7	8
Min Est Pct	50.2	31.1	8.0	0.0
Med Est Pct	53.7	36.0	10.1	0.3
Max Est Pct	56.8	38.3	12.2	4.7

BIAS OF ANIMAL POPULATION TREND ESTIMATES

Paul H. Geissler and William A. Link, U.S. Fish and Wildlife Service, Patuxent Wildlife Research Center

Surveys of animal calls or signs are often used to monitor population levels (Seber 1982, Ralph and Scott 1981). For example, the Mourning Dove Call-Count Survey (Dolton 1977) is a stratified random survey with more than 1000 routes. Each spring biologists count the number of doves they hear calling under standardized conditions at 20 stops along each route. The routes are used each year without drawing a new sample.

Biologists want to know if the animal population is increasing or decreasing over some period. Annual mean counts per route cannot be used because changes in routes and observers affect the counts. Instead, the slope of a regression line is used to estimate the average trend on each route over the period of interest and to predict counts in years $\tilde{y}+1$ and $\tilde{y}$. These trends are used to estimate the ratio of the populations in those years (Geissler 1984).

The call-counts on a route can be modeled as

$$c_{sriy} = \theta_{sri}\, \tau_{sr}^{y}\, \epsilon_{sriy} \qquad (1)$$

where c_{sriy} = observed call-count, s = stratum, r = route, i = observer, y = year, θ_{sri} = observer effect, τ_{sr} = the trend, and ϵ_{sriy} = error term. Taking logarithms, (1) becomes a linear regression.

$$c''_{sriy} = \theta''_{s\cdot i} + \tau''_{sr}\, y + \epsilon''_{sriy} \qquad (2)$$

where $c''_{sriy} = \ln(c_{sriy}+0.5)$. Quantities on the logarithmic scale are indicated by a double prime to distinguish them from the corresponding quantities on the arithmetic scale. Because the logarithm of zero is not defined, an arbitrary positive constant is added to c_{sriy} (0.5 is halfway between zero and the lowest observable count).

The adjusted predicted count in year y is

$$\tilde{c}''_{sry} = \bar{\theta}''_{sr} + \hat{\tau}''_{sr}\, y \qquad (3)$$

where $\bar{\theta}''_{sr}$ is the mean of the estimated observer effects on the route r (see Searle 1971 Ch. 5, Searle, et al. 1980). Ordinary linear regression provides the best linear unbiased estimate of $\tilde{c}''_{sry}$ from (3). Suppose that ϕ'' is an estimable linear combination of the regression parameters on the logarithmic scale. Under the assumption that ϵ''_{sry} is normally distributed (counts are lognormally distributed), Bradu and Mundlak (1970) have shown that the uniformly minimum variance unbiased estimate (UMVUE) of $\phi = \exp(\phi'')$ is

$$\phi = T(\hat{\phi}'') = \exp(\hat{\phi}'')\, g_f\Big(-\frac{f+1}{2f}\, v \Big) \qquad (4)$$

where $v = \nu(\hat{\phi}'')$, f = error degrees of freedom, and the function g_f is defined by

$$g_f(t) = \sum_{p=0}^{\infty} \frac{f^p\,(i+2p)}{f\,(f+2)\cdots(f+2p)} \left(\frac{f}{f+1}\right)^p \frac{t^p}{p!} \quad .$$

In particular, $\tilde{c}_{sry} = T(\tilde{c}''_{sry})$.

The population trend can now be estimated as the ratio of the total populations in years $\tilde{y}+1$ and $\tilde{y}$ based on a sample of n_s out of the N_s sampling units in stratum s. Assuming that the predicted count $\tilde{c}_{sry}$ is proportional to the local population $\hat{P}_{sry} = a\, k\, \tilde{c}_{sry}$, where a is the area of the sampling unit and k is a proportionality constant,

$$\hat{\tau} = \frac{\sum_s \sum_r^{N_s} \hat{P}_{sr(\tilde{y}+1)}}{\sum_s \sum_r^{N_s} \hat{P}_{sr\tilde{y}}} \quad .$$

$$\text{Thus } \hat{\tau} = \frac{\sum_s A_s \sum_r^{n_s} \tilde{c}_{sr(\tilde{y}+1)} / n_s}{\sum_s A_s \sum_r^{n_s} \tilde{c}_{sr\tilde{y}} / n_s} = \frac{\sum_s A_s \sum_r^{n_s} \tilde{c}_{sr\tilde{y}}\, \hat{\tau}_{sr} / n_s}{\sum_s A_s \sum_r^{n_s} \tilde{c}_{sr\tilde{y}} / n_s} \quad . \qquad (5)$$

The back transformed adjusted counts $\tilde{c}_{sry}$ from (3) and (4) are weighted by the strata areas A_s, where $A_s = a\, N_s$. They may also be weighted by the inverse of the relative variance $w_{sr\tilde{y}}$ to increase the precision of the trend estimate, giving more weight to routes that have the smallest relative variance.

$$\hat{\tau} = \frac{\sum_s A_s \sum_r \tilde{c}_{sr(\tilde{y}+1)}\, w_{sr\tilde{y}}}{\sum_s A_s \sum_r \tilde{c}_{sr\tilde{y}}\, w_{sr\tilde{y}}} = \frac{\sum_s A_s \sum_r \tilde{c}_{sr\tilde{y}}\, \hat{\tau}_{sr}\, w_{sr\tilde{y}}}{\sum_s A_s \sum_r \tilde{c}_{sr\tilde{y}}\, w_{sr\tilde{y}}} \qquad (6)$$

Here the weight $w_{sr\tilde{y}} = \hat{\nu}(\epsilon''_{sry}) \,/\, \hat{\nu}(\tilde{c}''_{s\cdot\tilde{y}})$. The weights for the mid year are used for both the numerator and denominator. Note that the weight depends only on known values (route, year, and observer) and not on the variance that is estimated poorly.

The route is the only randomly selected element in the sampling design. Counts are repeatedly made on the same routes without selecting a new sample. Therefore variances should be calculated among routes rather than among years.

Bootstrap confidence intervals (Efron 1982) are estimated for the trend estimates (6). $\tilde{c}_{sr(\tilde{y}+1)}$, $\tilde{c}_{sr\tilde{y}}$, and $w_{sr\tilde{y}}$ are estimated for each route. A large number, B, of bootstrap samples each with n_s routes are selected with replacement from the n_s routes in each stratum and B bootstrap replicate estimates are made for a state or management unit using the parameter estimates for the selected routes. The state or management unit trend is estimated as $\hat{\tau}$ and the 100 α percent confidence intervals is estimated as $\hat{\tau} \pm \hat{\sigma}\, t_{\alpha,n-L}$, where $\hat{\tau}$ and $\hat{\sigma}$ are the mean and standard deviation of the bootstrap samples, and where t = tabulated t value, $n = \Sigma\, n_s$ is the total number of routes, and L is the number of strata. The bootstrap trend estimate $\hat{\tau}$ is reported to reduce the bias of a ratio from order n^{-1} to order n^{-2} (Efron 1982). B=200 bootstrap replications is recommended to give an adequate approximation.

In this paper we examine the accuracy and precision of the trend estimator. In the first section, we report the results of simulations which investigate the performance of the estimator under a variety of conditions. The second section investigates the performance of alternative estimators based on reduced Mean Squared Error (MSE) estimators. These alternative estimators are investigated because of the inadmissibility of the Bradu and Mundlak UMVUE (see, for example, Rukhin 1986).

We thank C. Bunck, J. Hatfield, C. McCulloch, and N. Coon for reviewing this manuscript.

I. SIMULATION

A factorial simulation experiment was performed to examine the bias and precision of alternate estimators using the GAUSS programming language (Edlefsen and Jones 1986). The factors (levels) were distributions (3 lognormals, Poisson, negative

binomial), trends (0.95, 1.00, 1.05), years (3, 5, 10), routes (10, 100), observers in model (yes, no), trend definitions $[P_{(\bar{y}+1)}/P_{\bar{y}}, P_{(\bar{y}+0.5)}/P_{(\bar{y}-0.5)}]$, and bias adjustment (no adjustment, mean of bootstrap replicates, median of bootstrap replicates). The estimators were developed using the lognormal model, but this distribution gives continuous counts without any zeros. Poisson and negative binomial counts test the trend estimation with more realistic discrete data with zero counts. The trends represent a stable, an increasing and a decreasing population of birds. Trends are estimated over several periods of time, ranging from 2 to 25 years. Two year trends are not included because a variance for the back transformation cannot be estimated. Varying the number of routes checks to see if bias is reduced with increased sample size. Observer effects may be very important, but including observers in the model may result in overparameterization. The trend τ is defined as the population in year $(\bar{y}+1)$ divided by the population in year $\bar{y}$ $[P_{(\bar{y}+1)}/P_{\bar{y}}]$. Alternatively it could be defined as the population in year $(\bar{y}+0.5)$ divided by the population in year $(\bar{y}-0.5)$ $[P_{(\bar{y}+0.5)}/P_{(\bar{y}-0.5)}]$. The alternate definition is centered on the period of interest but introduces another parameter into the denominator of the ratio ($\hat{c}^{m}_{sr\bar{y}} = \hat{\theta}^{m}_{sr}$ and $\hat{c}^{m}_{sr(\bar{y}-0.5)} = \hat{\theta}^{m}_{sr} - 0.5\ \hat{\tau}^{m}_{sr}$). The effectiveness of the bootstrap bias adjustment is evaluated. The mean of the bootstrap replicates reduces the bias from order n^{-1} to order n^{-2}. If the bootstrap empirical distribution function formed by the replicates is asymmetrical, the median may be a better estimator.

For the lognormal simulations, log counts were sampled from a normal distribution with mean $\bar{c}^{m}$ of 3.0 and standard deviations s^{m} of 0.1, 0.5, and 1.0. Mourning dove call-counts have a mean of about 20 birds per route ($\ln(20) \approx 3$) and the log transformed residuals have a standard deviation of about 0.5. Poisson counts with a mean of 2 birds per route and negative binomial counts with a mean of 0.3 birds per route and shape parameter k=0.5 were also used. We have found that American woodcock counts in low density areas are approximately distributed according to that negative binomial distribution and represent an extreme situation. A constant (0.5) was added to the Poisson and negative binomial counts so that zero counts could be log transformed (zero counts cannot occur with lognormal counts).

The specified mean count $\bar{c}$ corresponds to the mean year $\bar{y}$. Years were coded so that $\bar{y}=0$. The means for other years y were $\bar{c}\tau^{y}$ where τ is the trend (0.95, 1.00, or 1.05). Each year there was a 0.2 probability that the observer changed on the route, though all simulated observer effects were identical. These effects were included to assess the consequence of estimating these additional parameters which are often required in practice. In the analysis, bias adjustment (none, mean, median) are repeated measures because they result from the same simulated data. With 10 routes, 2,000 replications were run for each case, but with 100 routes, 500 replications per case were used.

The results of an analysis of variance of the factorial simulation experiment had many significant interactions. Only the bias adjustment and estimator effects are under the control of the investigator. The $[P_{(\bar{y}+1)}/P_{\bar{y}}]$ estimator was uniformly less biased than the $[P_{(\bar{y}+0.5)}/P_{(\bar{y}-0.5)}]$ estimator and will be adopted. Both bias adjustments had small effects; neither was consistently superior. Mean biases were 0.00364 with the median adjustment, 0.00546 with the mean adjustment, and 0.00586 without adjustment. Because these differences were small and inconsistent, the common practice of using the mean adjustment will be adopted.

Because of the significant interactions, each distribution was analyzed separately (Tables 1 and 2). Biases for the lognormal distribution with $s^{m}=0.1$ were negligible. With $s^{m}=0.5$ and 1.0, fitting observer effects with 3 or 5 years is not advisable because of the large biases. There may be too few degrees of freedom to obtain a stable variance estimate for the Bradu and Mundlak (1970) backtransformation. Otherwise, the biases seem to be acceptably small. The same recommendations apply to the Poisson distribution with the addition that 3 years may result in unacceptable biases. Ten year trends and 5 year trends without observer effects have acceptable biases. The negative binomial distribution represents an extreme situation with a mean count of 0.3 birds per route. Biases for that distribution are unacceptable. Adding 0.5 to data sets with numerous zeros biases the trends towards 1.0.

The standard errors of the trend estimates are given in Table 3. Increasing the variance of the counts of course increased the standard error of the estimate. Increasing the number of years or routes reduced the standard errors of the estimates as did not fitting observer effect which increased the effective number of years.

II. REDUCED MSE ESTIMATION OF e^{μ}

Considerable attention has focused on the inadmissibility of Bradu and Mundlak's (1970) estimator (Teekens and Koerts 1972, Evans and Shaban 1974, Rukhin 1986). Inadmissibility results from the possibility that $\exp(\bar{y})\cdot g_m(-.5s^2)$, which estimates a nonnegative quantity, can be negative. It is therefore possible to construct estimators which, though biased, have smaller mean squared errors (MSE) than Bradu and Mundlak's UMVUE.

Teekens and Koerts (1972) demonstrated that for any given m there exist negative values of t for which $g_m(t)$ is negative. As an illustration of this, consider the case m=1, which arises in the problem under consideration when trends are being estimated for a three year period. It is easily verified by consideration of the Taylor series for the cosine, that for $t \le 0$, $g_1(t) = \cos\left(\sqrt{-t}\right)$. Thus

$$P\left(\exp(\bar{y})\cdot g_1(-.5s^2) < 0\right) = P\left(\cos\left(s/\sqrt{2}\right) < 0\right)$$

$$= P\left((4j+1)\frac{\pi}{2} < \frac{s}{\sqrt{2}} < (4j+3)\frac{\pi}{2}\ ;\ \text{some } j \in (0,1,2,\cdots)\right), \qquad (7)$$

so that, since s^2/σ^2 has a chi-square distribution with one degree of freedom and cumulative distribution denoted by $\chi^2_1(\cdot)$, we find that the probability of a negative estimate is

$$\sum_{j=0}^{\infty}\left\{\chi^2_1\left(\frac{(4j+1)^2\pi^2}{2\sigma^2}\right) - \chi^2_1\left(\frac{(4j+3)^2\pi^2}{2\sigma^2}\right)\right\}.$$

Some values of this are given in the following table. For small values of σ the probability of a negative estimate is not too large, but as σ increases, the probability approaches 1/2.

σ	P(neg)	σ	P(neg)
0.50	.000001	1.75	0.2042
0.75	0.0031	2.00	0.2658
1.00	0.0263	2.25	0.3204
1.25	0.0754	2.50	0.3666
1.50	0.1386	2.75	0.4039

Because of the potential of negative estimates of trend for individual routes and the combined trend ratio estimator's bias might be attributable to large variability in the denominator, we considered alternative estimators which utilized reduced MSE estimators in place of the UMVUE.

Let X and Y be independent random variables, $X \sim \mathcal{N}(\mu,\sigma^2)$. We consider estimators of the form

$$T(X,Y) = Y\, e^{X} \tag{8}$$

for the parameter $\theta = e^{\mu}$. Evans and Shaban (1976) express the MSE of estimators of this form by

$$MSE\Big(T(X,Y) \Big) = E\Big(Y\, e^{X} - \theta \Big)^2$$

$$= E\Big(e^{2X} \Big)\, E\left\{ Y - \frac{\theta\, E\Big(e^{X} \Big)}{E\Big(e^{2X} \Big)} \right\}^2 - \theta^2 \frac{\Big(E(e^{X}) \Big)^2}{E(e^{2X})} + \theta^2. \tag{9}$$

This latter equality is established by first expanding the binomial and then completing the square. Since

$$E\Big(e^{aX} \Big) = \exp\left\{ a\mu + \tfrac{1}{2} a^2 \sigma^2 \right\} ,$$

we have

$$\frac{\theta\, E\Big(e^{X} \Big)}{E\Big(e^{2X} \Big)} = \frac{\exp\Big\{ 2\mu + \frac{1}{2}\sigma^2 \Big\}}{\exp\Big\{ 2\mu + 2\sigma^2 \Big\}} = \exp\Big\{ -\tfrac{3}{2}\sigma^2 \Big\} ,$$

which would suggest that the choice of Y to minimize (9) should be an estimator of $\exp\Big\{ -1.5\, \sigma^2 \Big\}$.

An estimator of the form $\hat{\mu} = exp\Big\{ \bar{y} - 1.5\, s^2 \Big\}$ is therefore attractive on two counts: 1) it has the potential of reducing MSE, and 2) it is non-negative. In order to evaluate the performance of such estimators, simulations of the trend estimator with 3, 5, and 7 years and 3, 5, and 7 routes were performed. The logarithms of the counts were normally distributed with standard deviations of 0.75, 1.25, and 1.75; and mean of zero. A stable population was simulated. Observer effects were not included in the simulation. Each combination was replicated 200 times. The results are summarized in Table 4.

As previously indicated, the bias of the estimator based on the UMVUE increased with the underlying standard deviation. This appears to be the case as well with the estimator based on the reduced MSE version, though the bias in the latter case is negative. Also, in the latter case, the magnitude of the bias did not decrease with increased number of routes or years; in fact, it appears to increase. While the standard deviation of the second trend estimator was generally smaller (not included in the table), the decrease in size could not offset the sizable bias. For these reasons we do not recommend the reduced MSE estimator.

III. CONCLUSIONS

- *The $P_{(\bar{y}+1)}/P_{\bar{y}}$ trend estimator should be used instead of the $P_{(\bar{y}+0.5)}/P_{(\bar{y}-0.5)}$ estimator because it is less biased, although it has the disadvantage of not being centered on the interval .
- *The effect of bootstrap bias adjustment was small and not consistent.
- *The bias increased with an increase in the variance of a lognormal distribution.
- *With lognormal [$\bar{c}$=3 ($\bar{c}\approx$20 birds), s^2>0.5] and Poisson ($\bar{c}$=2 birds) counts, fitting observer effects with less than 5 annual observations is not recommended because of the bias. If observer effects are believed to be important, trends should not be estimated.
- *With Poisson ($\bar{c}$=2 birds) counts, it is not advisable to fit trends with less than 5 annual observations.
- *Negative binomial ($\bar{c}$=0.3 birds, k=0.5) counts represent an extreme situation where trend estimates are not advised.
- *Otherwise, the bias of estimated trends seems to be acceptably small in the situations considered.
- *Adding a constant to the Poisson and negative binomial counts biased the trend estimates towards one but is necessary because the logarithm of zero is not defined.
- *The reduced MSE backtransformation is not recommended because the bias does not decrease with an increase in sample size.

REFERENCES

Bradu, D. and Y. Mundlak. (1970), "Estimation in lognormal linear models," *J. Am. Stat. Assoc.* 65:198-211.

Dolton, D. D. (1977), *Mourning dove status report.* 1976. Spec. Sci. Rep.- Wildl. No. 208. U. S. Fish & Wildl. Serv., Washington, D. C. 27 p.

Edlefsen, L. E. and S. D. Jones. (1986), *GAUSS programming language manual.* Aptech Systems, Inc., Kent, WA. 466 p.

Efron, B. (1982), *The jackknife, the bootstrap and other resampling plans.* Society for Industrial Applied Mathematics, Philadelphia. 92 p.

Evans, I.G., and S.A. Shaban. (1974), "A note on estimation in lognormal models," *J. Am. Stat. Assoc.* 64:632-636.

Geissler, P. H. (1984), "Estimation of animal population trends and annual indices from a survey of call-counts or other indications," *Proceedings of the American Statistical Association, Section on Survey Research Methods p.* 472-477.

Ralph, C. J., and J. M. Scott (eds.) (1981), *Estimating numbers of terrestrial birds.* Studies in Avian Biology No. 6, Cooper Ornithological Society, Lawrence, KS. 630 p.

Rukhin, A.L. (1986), "Improved estimation in lognormal models," *J. Am. Stat. Assoc.* 81:1046-1049.

Searle, S. R. (1971), *Linear models.* Wiley, New York. 532 p.

Searle, S. R., F. M. Speed, and G. A. Milliken. (1980), "Population marginal means in the linear model: an alternative to least squares means," *American Statistician* 34:216-221.

Seber, G. A. F. (1982), *The estimation of animal abundance and related parameters.* Macmillan, New York. 654 p.

Teekens, R., and J. Koerts. (1972), "Some statistical implications of the log transformation of multiplicative models," *Econometrica* 40:793-819.

Table 1. Significance levels (P) for effects in separate analyses of variances of biases in simulation experiment for each count distribution. Values of P<0.05 and P<0.01 are flagged by + and *, respectively.

Effect	Lnor 0.1	Lnor 0.5	Lnor 1.0	Poisson	N.Binom.
Trend	0.0423 +	0.0825	0.0240 +	0.9724	0.0001 *
Years	0.0702	0.8513	0.2374	0.0007 *	0.0001 *
Routes	0.6291	0.7192	0.4201	0.3694	0.3558
Obs.	0.1591	0.2708	0.0172 +	0.1329	0.0003 *
T x Y	0.0433 +	0.0356 +	0.0218 +	0.1482	0.0103 +
T x R	0.4004	0.9588	0.9731	0.9552	0.9600
T x O	0.0836	0.0257 +	0.0341 +	0.0034 *	0.0131 +
Y x R	0.7876	0.8926	0.6181	0.6206	0.9005
Y x O	0.4985	0.9695	0.2257	0.2520	0.0252 +
R x O	0.6291	0.6573	0.5104	0.7214	0.3791

Table 2. Mean biases of trend estimates from simulation experiment. Effects of the number of routes are not included in this table because they were not significant (Table 1).

Lognormal
log mean=3.0 (~20 birds), log standard deviation = 0.1

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	-0.0010	-0.0005	0.0000	0.0000	0.0000	0.0000
1.00	0.0010	0.0000	0.0000	0.0000	0.0000	0.0000
1.05	-0.0010	0.0000	0.0000	0.0000	0.0000	0.0000

Lognormal
log mean=3.0 (20 birds), log standard deviation = 0.5

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	-0.0120	0.0015	0.0085	0.0010	0.0040	0.0000
1.00	0.0345	0.0000	0.0080	-0.0010	0.0010	0.0000
1.05	-0.0060	0.0020	0.0000	0.0025	0.0020	0.0000

Lognormal
log mean=3.0 (20 birds), log standard deviation = 1.0

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	-0.0010	0.0060	0.0365	-0.0015	0.0115	0.0005
1.00	0.1920	0.0095	0.0440	0.0015	0.0020	-0.0005
1.05	0.0065	0.0025	0.0030	0.0005	0.0105	0.0005

Poisson
mean=2.0 (birds)

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	-0.0380	-0.0090	-0.0225	0.0065	0.0030	0.0060
1.00	-0.0270	-0.0150	-0.0050	-0.0025	-0.0025	-0.0005
1.05	0.0165	0.0290	-0.0115	-0.0080	-0.0060	-0.0060

Negative Binomial
mean=0.3 (birds)

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	0.0680	0.0570	0.0440	0.0065	0.0445	0.0370
1.00	0.0740	0.0285	0.0305	0.0030	-0.0020	0.0005
1.05	-0.0005	-0.0110	-0.0375	-0.0330	-0.0305	-0.0375

Table 3. Standard Errors of trend estimates from simulation experiment.

Lognormal
log mean=3.0 (20 birds), log standard deviation = 0.1

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	0.0000	0.0005	0.0000	0.0000	0.0000	0.0000
1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.05	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Lognormal
log mean=3.0 (20 birds), log standard deviation = 0.5

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	0.0020	0.0015	0.0005	0.0000	0.0000	0.0000
1.00	0.0005	0.0030	0.0010	0.0000	0.0000	0.0000
1.05	0.0000	0.0010	0.0020	0.0015	0.0000	0.0000

Lognormal
log mean=3.0 (20 birds), log standard deviation = 1.0

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	0.0130	0.0020	0.0015	0.0005	0.0015	0.0005
1.00	0.0460	0.0015	0.0400	0.0015	0.0010	0.0005
1.05	0.0085	0.0005	0.0080	0.0015	0.0005	0.0005

Poisson
mean=2.0 (birds)

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	0.0030	0.0010	0.0005	0.0005	0.0010	0.0000
1.00	0.0000	0.0020	0.0010	0.0005	0.0005	0.0005
1.05	0.0045	0.0030	0.0005	0.0010	0.0000	0.0000

Negative Binomial
mean=0.3 (birds)

	years(observer)					
trend	3(y)	3(n)	5(y)	5(n)	10(y)	10(n)
0.95	0.0040	0.0010	0.0010	0.0005	0.0005	0.0000
1.00	0.0020	0.0025	0.0085	0.0000	0.0010	0.0005
1.05	0.0065	0.0010	0.0005	0.0010	0.0005	0.0005

Table 4. Bias of trend estimators using UMVUE and reduced MSE estimators of e^{μ}.

			UMVUE		REDUCED MSE	
Years	Routes	S.D.	Bias	SE(Bias)	Bias	SE(Bias)
3	3	0.75	0.103	0.029	-0.084	0.029
		1.25	0.096	0.065	-0.108	0.056
		1.75	0.373	0.126	0.215	0.104
	5	0.75	0.032	0.018	-0.121	0.019
		1.25	0.231	0.057	0.011	0.051
		1.75	0.329	0.141	0.344	0.138
	7	0.75	0.083	0.017	-0.060	0.017
		1.25	0.239	0.050	0.059	0.048
		1.75	0.629	0.215	0.508	0.145
5	3	0.75	0.006	0.010	-0.050	0.010
		1.25	0.022	0.021	-0.092	0.020
		1.75	0.020	0.029	-0.168	0.029
	5	0.75	0.012	0.008	-0.042	0.008
		1.25	-0.006	0.016	-0.111	0.015
		1.75	0.060	0.028	-0.098	0.030
	7	0.75	0.002	0.007	-0.049	0.007
		1.25	0.038	0.014	-0.080	0.013
		1.75	0.010	0.021	-0.138	0.021
7	3	0.75	-0.003	0.006	-0.023	0.006
		1.25	0.004	0.011	-0.046	0.011
		1.75	0.003	0.015	-0.087	0.014
	5	0.75	-0.003	0.005	-0.024	0.005
		1.25	0.009	0.008	-0.040	0.008
		1.75	0.028	0.013	-0.062	0.013
	7	0.75	-0.005	0.004	-0.026	0.004
		1.25	0.015	0.007	-0.035	0.007
		1.75	0.019	0.011	-0.069	0.011

The Elimination of Quantization Bias using Dither

Douglas M. Dreher and Martin J. Garbo, Hughes Aircraft Company

I. Introduction

This paper presents a method for recovering the decimal precision of a non-observable variable that has been quantized. The technique involves adding a random variate (dither) from a uniform distribution to the variable prior to quantization. It then shows the conditions under which the expectation of the dithered quantization function equals the value of the variable in question. An expression for the variance of the dithered quantization function is also derived. The results are generalized to the multiple-quantization case. Examples are presented which show the application of this technique to reduce the magnitude of bias error caused by roundoff.

II. Methodology

Suppose that it is desired to estimate the actual value of a variable, F, when it can be observed only after being quantized. Specifically, if q is the quantization interval, or distance between successive quantum levels, then F may be represented as

$$F = nq + z \tag{1}$$

where n is an integer and $|z| < q$. If the quantization interval is 1, then z is the fractional part of F. Thus, the problem can be reduced with no loss of generality to that of estimating z given its quantized value, $Q(z)$.

Assume that the non-observable variable z is quantized as follows:

$$Q(z) = \mathrm{INT}[z + 0.5] \tag{2}$$

where $Q(z)$ is the quantized value of z and $\mathrm{INT}[x] \equiv$ largest integer $\leq x$. Fig. 1 illustrates this quantization function.

Furthermore, define a dither density function $f(x)$ to be a uniform density function such that

$$f(x) = \begin{cases} 1, & z - 0.5 \leq x < z + 0.5 \\ 0, & \text{elsewhere} \end{cases} \qquad -1 < z < 1. \tag{3}$$

When a random variable x from this density is added to z prior to quantization it then follows that the expectation of $Q(z)$, $\mu_Q(z)$, may be expressed as

$$\mu_Q(z) = \int_{-\infty}^{\infty} Q(x) f(x)\,dx \tag{4a}$$

$$= \int_{z-0.5}^{z+0.5} Q(x)\,dx \tag{4b}$$

$$= z \tag{4c}$$

and the variance of $Q(z)$, $\sigma_Q^2(z)$, as

$$\sigma_Q^2(z) = \int_{-\infty}^{\infty} [Q(x)]^2 f(x)\,dx - \mu_Q^2(z) \tag{5a}$$

$$= \int_{z-0.5}^{z+0.5} [Q(x)]^2\,dx - z^2 \tag{5b}$$

$$= |z|(1 - |z|). \tag{5c}$$

The simplified expressions for $\mu_Q(z)$ and $\sigma_Q(z)$ result from the method of quantization. In particular, (2) generates equal-size steps symmetric about the origin as shown in Fig. 1.

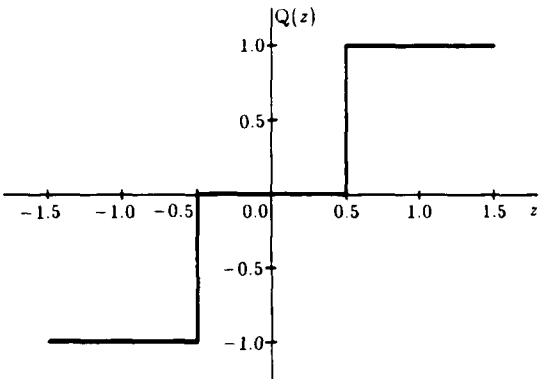


Figure 1. Symmetric equal-step quantization.

It is conceivable that a variable might undergo multiple quantizations prior to its utilization. In such cases the question arises as to the extent to which dither should be applied. For example, a variable could be quantized to quarter-unit precision and then be rounded via (2) to unit precision. This process results in the quantization diagram of Fig. 2. It should be noted that this diagram is equivalent to that of Fig. 1 shifted to the left by 0.125. This shift, or bias, is

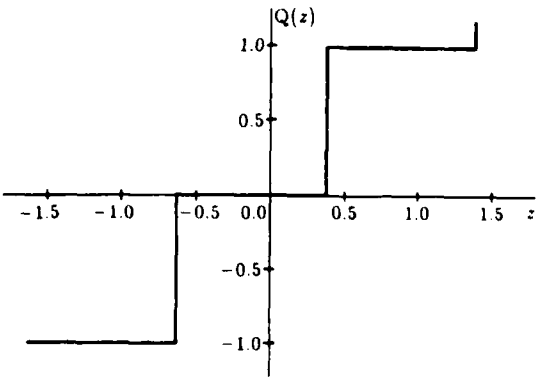


Figure 2. Biased equal-step quantization.

caused by the upward rounding of values midway between quantization intervals; in this case, the rounding of 0.125 to 0.25, etc. Applying (3) and (4) to this quantization function results in

$$\mu_Q(z) = z+0.125. \tag{6}$$

The bias in (6) is eliminated by redefining the dither density function as

$$f(x) = \begin{cases} 1, & z-0.625 \leqslant x < z+0.375 \\ 0, & \text{elsewhere} \end{cases} \qquad -1<z<1.$$

This bias can also be eliminated by applying a dither with density function

$$f(x) = \begin{cases} 4, & z-0.125 \leqslant x < z+0.125 \\ 0, & \text{elsewhere} \end{cases} \qquad -1<z<1$$

prior to quarter-unit quantization and then applying a dither with density function

$$f(x) = \begin{cases} 1, & z-0.5 \leqslant x < z+0.5 \\ 0, & \text{elsewhere} \end{cases} \qquad -1<z<1$$

prior to unit quantization. Either method results in an unbiased estimation of z.

Any unbiased estimator of z will have an associated discrete probability density function. For $-1<z<0$ the discrete values will be -1 and 0. For $0<z<1$ the values will be 0 and 1. In either case the variance of this function depends on z as given in (5c). Fig. 3 plots the standard deviation of the density function for $q = 1$. This plot, which consists of half-circles symmetric about the error axis, shows the error increasing from zero at each quantum level to 0.5 halfway between quantum levels.

The application of dither may be generalized to n successive quantizations with quantization intervals $q_1, q_2, \cdots, q_n$. If a dither is applied before each quantization then the respective dither density functions are

$$U\left[z-\tfrac{1}{2}q_i,\ z+\tfrac{1}{2}q_i\right] = \begin{cases} q_i^{-1}, & z-\tfrac{1}{2}q_i \leqslant x < z+\tfrac{1}{2}q_i \\ 0, & \text{elsewhere} \end{cases}$$

for $i = 1, 2, \cdots, n$, where $-q_n<z<q_n$. This gives an unbiased estimator $\mu_Q(z)$ with variance given by (5c) with $q = q_n$. However, a single dither can be used before the first quantization if the introduced bias can be removed. For n quantizations with a single dither from $U\left[z-\tfrac{1}{2}q_n,\ z+\tfrac{1}{2}q_n\right]$, the resulting bias is

$$\tfrac{1}{2}(q_1+q_2+\cdots+q_{n-1}).$$

To remove this bias the dither density function is redefined as

$$U\left[z-\tfrac{1}{2}q_n-\tfrac{1}{2}(q_1+q_2+\cdots+q_{n-1}),\right.$$
$$\left. z+\tfrac{1}{2}q_n-\tfrac{1}{2}(q_1+q_2+\cdots+q_{n-1})\right].$$

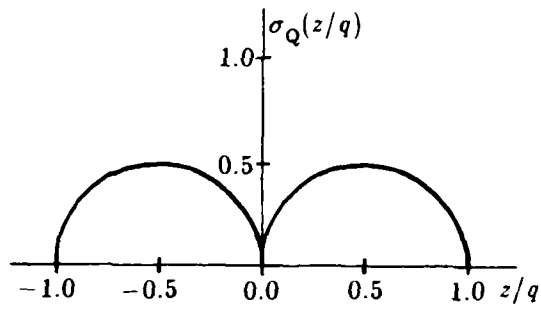


Figure 3. Quantization error standard deviation.

A dither with this density function is referred to as *consolidated dither*. With the bias removed we now have an unbiased estimator with variance given by (5c) with $q = q_n$.

Depending on the method of rounding used, the application of dither can introduce different biases when negative numbers or numbers close to zero are involved. A discussion of various rounding methods, introduced biases, and methods for avoiding these biases is included in an appendix.

III. Application

Suppose we have two computers, A and B, where A is generating a series of numbers which are then sent to B. The numbers in the series are floating-point which, for simplicity, are assumed to be restricted to the interval [0,1] with 6 significant digits of precision. Also, for simplicity, it is assumed that during the time of interest each of the numbers is equal to a constant. This is not a requirement for the method to work, but makes it easier to grasp the concepts involved.

Due to the limited memory and processing speed of computer B, it cannot handle the series of numbers in its original floating-point format, but can handle only integers. Therefore, computer A rounds each number to the nearest integer (0 or 1) before sending it to B. B accumulates the series of numbers and reports the running sum after each number is added. Since all of the numbers in the series are the same and are therefore rounded to the same integer, either 0 or 1, a constant error is introduced. This produces a bias which is accumulated by B and results in an error in the sum which increases with each number added. This is unacceptable, but if computer B cannot be upgraded or replaced, then only integers can be sent from A to B. Fortunately, there is a way to remove the bias in the accumulated sum, at the expense of introducing some variance in the value of the sum: we add a random dither to the numbers before rounding them. Since integer rounding is used, the dither should be from $U[z-0.5, z+0.5]$, where z is the current number to be rounded.

Fig. 4 shows the actual and true (no rounding) accumulated sums for a series of one hundred numbers, each equal to 0.5 and rounded up to 1. Fig. 5 shows the same series with dither applied before

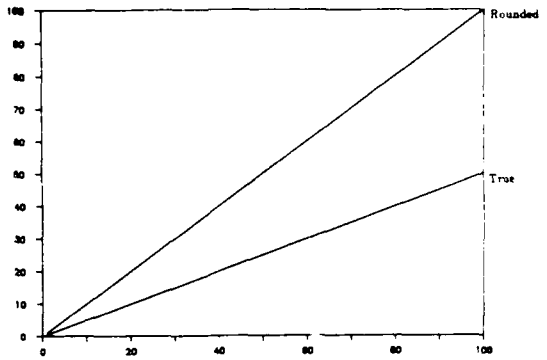


Figure 4. Accumulated sum of a series of 100 numbers, each equal to 0.5.

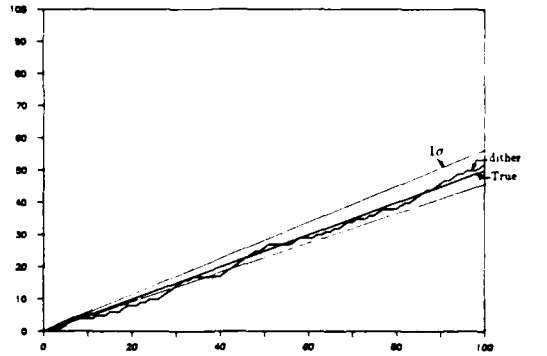


Figure 5. Accumulated sum of a series of 100 numbers, each equal to 0.5 plus a dither generated from $U[z-0.5,\ z+0.5]$.

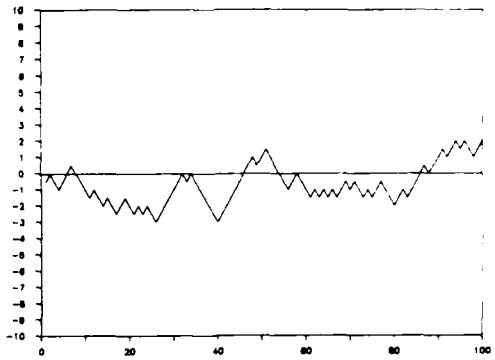


Figure 6. Accumulated error of the sum of Fig. 5.

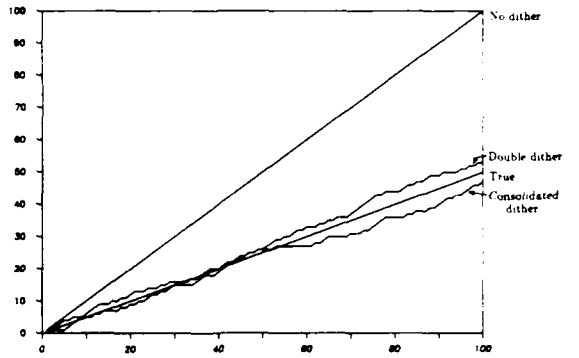


Figure 7. Accumulated sums with no dither, double dither, and consolidated dither applied to a series of 100 numbers, each consisting of 0.5, rounded to the nearest multiple of 0.25, then rounded to the nearest integer.

rounding, and Fig. 6 shows the accumulated error. A marked improvement can be seen when dither is added.

Now we'll complicate the problem by adding another computer, C, between A and B. Computer C can handle numbers with two fractional bits of precision; therefore, A rounds numbers to the nearest multiple of 0.25 before sending them to C, which in turn rounds them to integers and sends them to B, which accumulates them as before. The accumulated sum will still have a bias that can be removed using dither. We could apply a $U[z-0.125,\ z+0.125]$ dither in A before rounding to the nearest multiple of 0.25, followed by a $U[z-0.5,\ z+0.5]$ dither applied in C before rounding to the nearest integer (*double dither*). However, in Section II it was shown that a single consolidated dither can be applied instead, with density function

$$U[z-0.5-0.125,\ z+0.5-0.125]$$
$$= U[z-0.625,\ z+0.375].$$

This dither is applied in A before rounding to the nearest multiple of 0.25. Fig. 7 shows the accumulated sums for no dither, double dither, and consolidated dither, along with the true sum. Both the double dither and consolidated dither remove the bias, but the consolidated dither requires less computation.

IV. Summary

This paper developed the application of dither to recover lost precision and reduce biases introduced by quantization. Both the simple case with one quantization of a variable and the case of multiple quantizations were considered. In all cases the dithering technique results in an unbiased estimate of an unobservable variable in addition to knowledge of the variance about the estimate. The methodology is illustrated in a computer communications application.

Appendix

Care must be taken when using dither to avoid introducing a bias when rounding numbers that are close to or less than zero. Dither requires equal size quantization intervals for unbiased results. Depending on the rounding method used, the quantization intervals may or may not be of equal size.

There are several methods of rounding likely to be used on computers, each rounding values to the nearest integer (integer rounding is assumed here for simplicity). They differ in the way they treat values with fractional part 0.5. The first method, which we will call *normal rounding*, is

$$Q(z) = \mathrm{sgn}(z)\cdot \mathrm{INT}[|z| + 0.5].$$

where

$$\mathrm{sgn}(x) = \begin{cases} -1, & x<0 \\ 0, & x=0 \\ 1, & x>0 \end{cases}$$

and

$$\mathrm{INT}[x] \triangleq \text{largest integer} \leqslant x.$$

Using this method, 0.5 is rounded to 1 while −0.5 is rounded to −1.

The second method, which we will call *upward rounding*, is

$$Q[z] = \mathrm{INT}\, z + 0.5 .$$

In this method, all values with fractional part 0.5 are rounded up to the next larger integer; thus, −0.5 is rounded to 0 instead of −1. As will be seen shortly, this is the most convenient form of rounding to use with dither when negative values will be encountered.

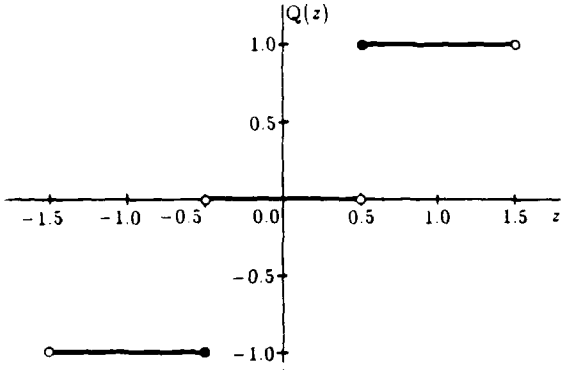
Fig. A1 shows the quantization intervals obtained by each method. Notice that upward rounding gives uniform quantization intervals $\{\cdots, [-1.5, -0.5), [-0.5, 0.5), [0.5, 1.5), \cdots\}$ whereas normal rounding has intervals $\{\cdots, (-1.5, -0.5], (-0.5, 0.5), [0.5, 1.5), \cdots\}$. While this difference between normal and upward rounding may seem insignificant, it becomes important when consolidated dither is used. Fig. A2 shows the expected values when a consolidated dither from $U[z-0.5, z+0.5]$ is applied before rounding to the nearest multiple of 0.25 and then rounding to the nearest integer. When upward rounding is used, there is a consistent bias of +0.125 which is easily removed by using a dither from $U[z-0.625, z+0.375]$. When normal rounding is used, there is a bias of +0.125 for positive values and −0.125 for negative values, separated by a transition region located between ±0.125. If a dither from $U[z-0.625, z+0.375]$ were used, then a bias of −0.25 would result for negative values. To remove any bias we can perform dithering as follows.

$$z_1 = \lfloor z \rfloor + 1 \tag{A1a}$$

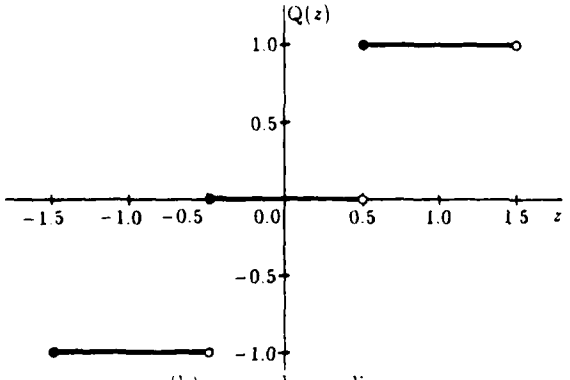
$$z_2 = U[z_1 - 0.625,\ z_1 + 0.375] \tag{A1b}$$

$$z_3 = \mathrm{sgn}(z) \cdot (\mathrm{INT}[z_2 + 0.5] - 1) \tag{A1c}$$

The addition of 1 in (A1a) is used to move the range of dithered values away from the transition region and is subtracted in (A1c).

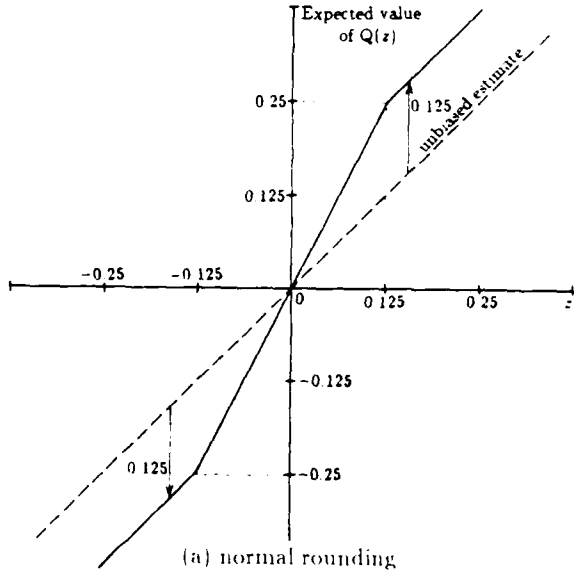


(a) normal rounding

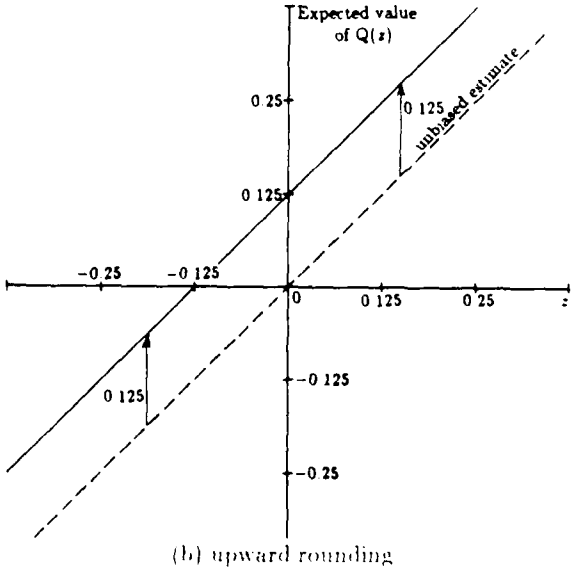


(b) upward rounding

Figure A1 Quantization intervals.



(a) normal rounding



(b) upward rounding

Figure A2 The bias introduced when applying a $U[z-0.5, z+0.5]$ dither before rounding to the nearest multiple of 0.25.

AN ALTERNATE METHODOLOGY FOR SUBJECT DATABASE PLANNING

Henry D. Crockett, Mark E. Eakin, and Craig W. Slinkman

ABSTRACT

An important aspect of data administration is strategic data planning. Strategic data planning is the scheme which an enterprise uses to ensure that its information systems function can support the managerial objectives of the enterprise. An important component of strategic data planning is the determination of the subject databases needed. James Martin has suggested a simple ad hoc procedure for performing this analysis. An alternative procedure is suggested using SAS to perform a multivariate statistic technique called correspondence analysis. This technique has the advantages that it has a strong theoretical justification, yields a numeric measure of the strength of the subjective database clustering, is well understood, and is relatively simple to include in CASE software.

INTRODUCTION

James Martin in his book, Strategic Data-Planning Methodologies, presents an organized method by which organizations can design their data resources to meet their long term information needs. A corporation determines what processes it must carry out in order to thrive in business, then they determine what data might be needed in order to support these processes. The process model that Martin develops, called an Enterprise Model, is quite similar to IBM's Strategic Business Plan. However, IBM's plan does not provide the detail and the organized method needed to make their plan fully useful. Martin uses his Enterprise Model to determine the data requirements of the organization, and to divide these data requirements into subject databases that contain all of the information needed about specific entities in the corporate environment. By this method he hopes to eliminate duplicate information and effort in programming and planning.

Once the Enterprise Model and subject databases are determined they must then be separated into operational subsystems for implementation on a piecemeal basis. Martin's methodology for accomplishing this seems to be inadequate, and open to multiple interpretation and errors. This paper will address these problems and present an alternate method for determining these operationalizable subsystems.

The Enterprise Model

The enterprise model is a top down view of all activities which need to be performed in order to have a functional organization. All organization activities can be described as processes carried out by different functional areas in the organization. The functional areas of an organization refer to the major areas of activities carried out by the organization, such as finance, production, sales, distribution, and accounting. Each functional area can then be divided into the processes which must be carried out in order to meet the needs of the organization. The functional areas and processes should be those needed to maintain the existence of the corporation. An example would be the functional area of finance which would need to carry out the processes: financial planning, budgeting, capital acquisition, funds management, and banking. This Enterprise Model when complete should represent a comprehensive model of the activities carried out by the organization. It should also be an understandable and useful tool in understanding the operation of the organization as a whole, and it should remain true as long as there is no dramatic fundamental change in the organization's statement of purpose. Once the functional areas and processes of an organization have been established then the data which is necessary to support them can be determined by contacting the department or group which performs each process. However, the identification of the functional areas and processes should be totally independent of the current structure of the organizational chart.

Subject Databases

James Martin has coined the term subject database to represent the logical view of all data collected about entities in the corporate domain. Many information systems have already incorporated this methodology informally by grouping all records that are needed presently for one entity into one database. The difference between the classical method that may contain the correct information and the subject database method is that the classical view collects all the data needed by the application that is presently under construction, while a subject database would be constructed to include all information that will be needed in the foreseeable future. These subject databases would be created by the interaction of the data requirements of the corporation mapped onto the enterprise model of the corporation. In order to accomplish this, data classes of information created by examining entities in the organizations environment would be cross referenced as to which data classes are used as input and output of processes in the Enterprise Model. By systematically considering data needed by the organization, instead of the conventional manner of merely collecting data for each application as needed, an overview of the

data needs of the organization as a whole can be obtained. Such an overview could be very profitable for the corporation in terms of programming effort and timeliness. This would mean that whenever a new application is created the information necessary should have already been included in the database, therefore the applications programmer should not need to create his own data in order to create a new application. This would greatly decrease the time and effort required for new applications. Subject databases also reduce the number of databases necessary for operation. However, most organizations do not deal with a large number of entities. If files are designed for specific application, then the number of files needed grows almost as fast as the number of applications. This proliferation leads to redundant data, update errors, and poor design of application programs.

Grouping Subject Databases Into Easily Implementable Subsystems

In order to implement these subject databases, Martin suggests that the subject databases be grouped into implementable subsystems. Then the subsystems which satisfies the immediate needs of the corporation should be implemented first, hopefully to produce a new application which is acknowledged to be much needed in the organization. The approach that James Martin uses to divide the databases into implementable subsystems is similar to IBM's Business Systems Planning methodology. Both approaches rely on manual methods of manipulation, and some areas are ill defined and open to multiple answers depending on interpretation of the person organizing the sequence of subject databases.

First the processes of the organization are ordered by the life-cycle approach. Most service and manufacturing organizations tend to have a four stage life cycle: planning, acquisition, stewardship, and disposal. The databases are then entered as columns and the intersections of processes and databases are designated by a 'U' if the process uses the data in that database, or by a 'C' if the data in that database is created by that process. In Martin's example this matrix appeared as Figure #1.

Martin then changes the order of the subject databases so that the'C's are ordered from the top left hand corner to the bottom right hand corner. However, the order of the processes are not changed. This reordering is done manually and is subject to multiple interpretations. There is no one correct way in which to order the columns', different analyst may agree on the processes and the subject data bases and disagree on the way to order the columns. This disagreement can have far reaching effects since this ordering of the matrix is then grouped into subsystems for implementation purposes.

These groupings of 'C's into implementable subsystems is done by inspection. The selection of clustering is judgmental, however, Martin suggests an affinity analysis as a follow up step. However, this affinity analysis is very rough and does not provide hard quidelines of delineation of borderline cases into seperate groupings. Even given that the groupings are more or less correct, other problems arise from this method. Some of the 'U's fall outside of the groupings and are therefore considered as exterior data flows from one subsystem to another. Therefore, even attempting to implement these subsystems in an orderly manner may prove very difficult for the new subsystems will sometimes have to share data with old systems. This will produce incompatibilities and lead to patching of data flows and more data redundancy in the system, rather than less. It is precisely this sort of complication that will lead to problems in the attempt to organize the corporate data into a TYPE III environment.

Another major problem can be seen in the implementation scheme shown in Figure #2.

This matrix not only shows the problem of 'U's exterior to the subsystems, there is also a 'C' which is incompatible with Martin's arrangement of 'C's on the diagonal. The processes Budget Planning and Sales Forecasting both help to create the subject database Budget and the process are so far removed from each other in the life cycle that Budget cannot be arranged in any way that will bring both 'C's onto the diagonal. Martin disregards this inconsistency by not even mentioning it specifically. The only reference made to the problem of two processes creating one subject database and thereby potentially leading to this problem is a short statement to the fact that these types of databases might be candidates to be split up into two databases thereby artificially alleviating the problems.

Canonical Correlation Analysis

Canonical correlation uses linear compounds to describe the dependencies between two sets of variables [Morrison, 1976]. Let X_1 denote the first set of variables in which there are r_1 variables and N observations. The second set, denoted by X_2, contain r_2 variables and N observation. In this paper, N is the number of process by subject databases relations which contain either a 'U' or a 'C'. The first set of variables consist of N observations or r_1 indicator variables:

x_{1ij} = 1 if observation i is from row j_1
= 0 otherwise.

$i=1,2,\ldots,N \qquad j=1,2,\ldots,r_1$

and the second set consists of N

observations of r_2 indicator variables:

x_{2ij} = 1 if observation is from row j_1
= 0 otherwise.

$i=1,2,\ldots,N \qquad j=1,2,\ldots,4_2$

The first step in canonical correlation combines X_1 and X_2 into a single matrix with N rows and r_1+r_2 columns and calculates the sample variance-covariance matrix S where:

$S = [s_{ij}]$ and

$x_{ij} = [Sx_{ki}*x_{kj} - (Sx_{ki})\ (Sx_{kj})/N]/(N-1)$

the variance-covariance matrix is then partitioned into four submatrices:

$$S = \begin{matrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{matrix}$$

where S_{11} and S_{22} are the variance-covariance matrices of X_1 and X_2, respectively, and S_{12} contains the covariances of the X_1 variables with the X_2 variables. These variance-covariance matrices are used to calculate the descriptive linear compounds.

Canonical correlation describes the dependencies of X_1 and X_2 by $d_i = a_i'*X_1$, and

$v_i = b_i'*X_2 \qquad i=1,2,\ldots,m \qquad m=\min(r,c),$

such that d_1 and v_1 have the highest correlation among all pairs of linear compounds, d_2 and v_2 have the highest correlation among all pair of linear compounds orthogonal (or uncorrelated) with the first two, etc. Each pair of linear compounds are uncorrelated with all others. In the situation being studied, only d_1 and v_1 need to be considered since the purpose of this study was to diagonalize the matrix by rearranging the rows and columns. The linear combinations that show the strongest correlation also give the best diagonalization.

The values of a_1 and b_1 can be found by solving the two simultaneous equations:

$(S_{12}S_{22}S_{12}' - lS_{11})a_1 = 0$

$(S_{12}S_{11}S_{12}' - lS_{22})b_1 = 0$

where l is the largest characteristic root or eigenvalue of the following equation:

$S_{12}S_{22}S_{12}' - lS_{11} = 0$ or

$S_{12}S_{22}S_{12}' - lS_{22} = 0$

Statistical packages are available which will quickly calculate the vectors: a_i and b_i $(i=1,2,\ldots,m)$. The results used in this paper used PROC CANCORR of SAS, the Statistical Analysis System.

After finding d_1 and v_1, the values of X_1 and X_2 are substituted into d_1 and v_1, respectively, to obtain the canonical scores. These scores are then ranked from 1 to r for the d_1 values and ranked from 1 to c for the v_1 values, tied scores received the same rank. Since there are only r_1 unique values of d_1 and r_2 unique values of v_1, these new ranks establish the new row and column position of each observation. These new positions rearrange the rows and columns of the old matrix and establish a new matrix showing the strongest possible diagonalization.

The problem with canonical analysis is the interpretability of the subsystems that are grouped together. This procedure will maximize correlations between sets, however, it does not provide a facility for interpreting the resulting dimensions of the subsystems arranged by correlation. Another cause for concern might be the sensitivity of the solution to the inclusion of further 'U's and 'C's into the matrix itself at some latter date. This has the possibility of radically changing that solution to the correlations. Therefore, the matrix should be defined as completely as possible before the use of this methodology. Other theoretical limitations include outliers, multicollinearity and singularity. However, the method by which the information matrix is developed seem to negate the ill effects of these problems.

Application of Canonical Analysis

Canonical analysis was performed upon Martin's original matrix using the canonical correlation procedures in SAS. Canonical row and column scores were obtained using PROC CANCOR and the matrix was then arranged in ascending order along both dimensions. The result matrix appears in Figure #3.

This method appears to provide for a much more regular appearance than merely rearranging the columns, and is much less open to multiple interpretation. If all parties agree to the processes and subject database assignments then this method provides a statistically defensible method of rearrangement that is reproducible. In order to group the new arrangement into operationalizable subsystems the canonical variates are clustered using a SAS procedure called PROC CLUSTER. This produces differing numbers of cluster and the R-square for each number of clusters. These can then be plotted on two axis and the number of clusters determined by observing a bend in the resulting curve or by deciding how many clusters are adequate and defensible. If several possible numbers of clusters meet all requirements then it may be possible to determine which set divides the matrix into the most easily implemented subsets of systems. Although this does seem to be rather arbitrary, the procedure PROC CLUSTER will automatically determine which

observations fall in which sets given a chosen number of sets.

This procedure was performed on the above matrix and the optimal number of clusters was determined to be nine. This produced clusters which contain all of the 'U's and 'C's in the original matrix. Therefore there is no date being shuffled from one subsystem to another. This will reduce the time and effort spent patching and implementing a new system. The subsystems are self contained with the only implementation problem being that each subsystem will not contain all of the 'C's that create the data that is used in its 'U's. The clustering of subsystems produced by PROC CLUSTER appears in Figure #4.

CONCLUSION

The major advantage of this system of rearrangement of processes and subject databases and clustering of the resulting arrangement is that it can be fully automated. Martin always stresses that any system that is of this size and complexity should be automated as much as possible in order that more analysis be accomplished. This methodology could be used in such a way as to be interactive. Thereby, when an Enterprise Model is completed the subsystems could be determined at the press of a button. This system of determining the subsystems is also more easily defended, less judgmental, and more open to reproduction.

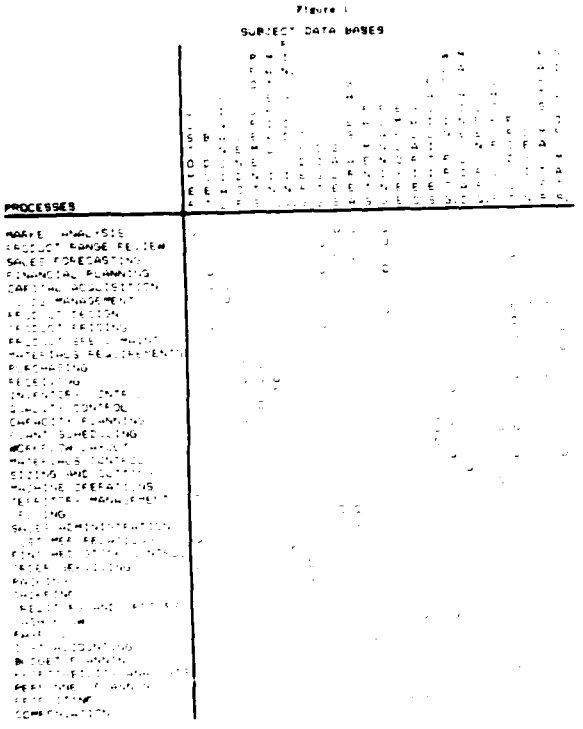
REFERENCES

1. Hair, Joseph F. Jr., R.E. Anderson, R.L. Tatham, B.J. Grablowski, Multivariate Data Analysis, Petroleum Publishing Co., 1979, p. 180-186.

2. Martin, J., Strategic Data-Planning Methodologies, Englewood Cliffs, N.J., Prentice Hall 1982.

3. Martin, J., Managing the Data-Base Environment, Englewood Cliffs, N.J., Prentice Hall 1983.

4. SAS User's Guide, SAS Institute, Inc., *1982 Edition.*



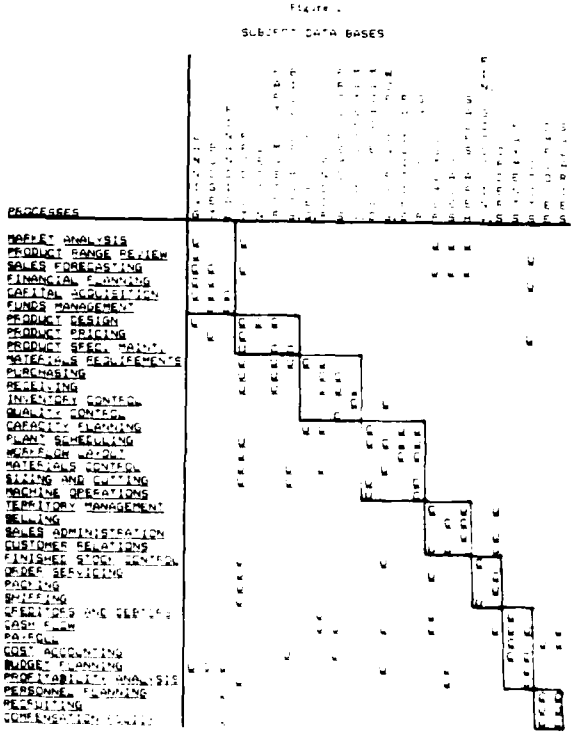
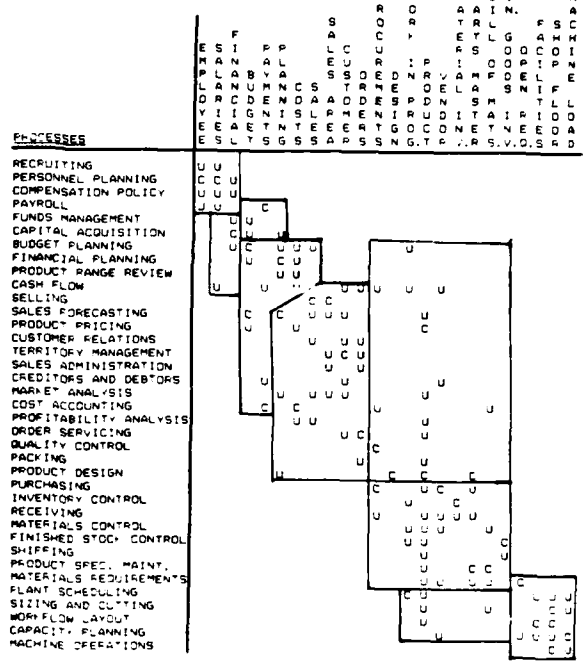


Figure 3
SUBJECT DATA BASES

PROCESSES	ESTIMATING	SALES ANALYSIS	FINANCIAL ACCOUNTING	BUDGET ANALYSIS	PAYROLL	PERSONNEL	SALES ORDER	PURCHASE ORDER	WORK ORDER	INVENTORY	MATERIALS REQUIREMENTS	BILL OF MATERIALS	FINISHED STOCK	SHIPPING	MACHINE OPERATIONS
RECRUITING	C					C									
PERSONNEL PLANNING	C			C		C									
COMPENSATION POLICY	C					C									
PAYROLL	C				C										
FUNDS MANAGEMENT			C												
CAPITAL ACQUISITION			C	C											
BUDGET PLANNING			C	C											
FINANCIAL PLANNING			C	C	C										
PRODUCT RANGE REVIEW					C										
CASH FLOW	C														
SELLING					C										
SALES FORECASTING		C			C										
PRODUCT PRICING		C													
CUSTOMER RELATIONS					C										
TERRITORY MANAGEMENT		C													
SALES ADMINISTRATION					C										
CREDITORS AND DEBTORS					C										
MARKET ANALYSIS		C													
COST ACCOUNTING			C												
PROFITABILITY ANALYSIS			C												
ORDER SERVICING							C								
QUALITY CONTROL									C						
PACKING									C						
PRODUCT DESIGN															
PURCHASING								C							
INVENTORY CONTROL										C					
RECEIVING								C							
MATERIALS CONTROL										C	C				
FINISHED STOCK CONTROL										C			C		
SHIPPING													C		
PRODUCT SPEC. MAINT.												C			
MATERIALS REQUIREMENTS											C	C			
PLANT SCHEDULING											C				C
SIZING AND CUTTING															C
WORKFLOW LAYOUT															C
CAPACITY PLANNING															C

Figure 4
SUBJECT DATA BASES



SENSITIVITY ANALYSIS OF THE HERFINDAHL-HIRSCHMAN INDEX

James R. Knaub, Jr., Energy Information Administration

Introduction:

The Herfindahl-Hirschman Index (HHI), the sum of the squares of the relative percent of sales made by each company in a market, has been used by the U.S. Government and industry to measure market concentration. A small HHI value indicates low concentration. One paper attributed to the U.S. Department of Justice (1982), suggests a value of 1000 for this index to delineate between "moderately concentrated" and "unconcentrated" markets. A value of 1000 could be the result of having ten companies, all with equal sales. However, if two of these companies were to merge, an HHI value of 1200 would result. Thus a twenty percent change in the HHI may be considered to be substantial in this case, or it could be considered to be a random change in a market, not indicative of a trend. If this index is calculated for different time periods for the same market, there is a question as to when one may say that a substantial change has taken place. If a small change in a frame often results in a large change in the HHI, then a small change in the HHI may not mean very much. Conversely, if a large change in a frame often results in a small change in the HHI, then one could say a small change in the HHI may be very important. (Note the similarity to Type I and Type II errors in classical hypothesis testing.)

Two approaches to the analysis of the sensitivity of this index are given in this paper. Both analyses are measured in terms of the coefficients of variation (cv), of simulated resulting HHI values when starting with a given market and allowing certain changes to take place in random fashions.

Description of Approaches:

The purpose of this paper is to discuss how one might determine whether a change in the HHI is substantial. Can we describe random changes in a market which are not indicative of a trend? In one approach, a bootstrap-like procedure was used to determine the variation in HHI values which could result should there be a replacement of actual data by data randomly chosen from the existing frame. The cvs obtained in this case seem an unfair judge of the performance of the HHI, however, since such a variety of possible sets of hypothetical "companies" contain perhaps too many scenarios which would not be considered comparable to the original, or observed scenario. "Restricted" case simulations here are those where only replications which resulted in a total sales volume within five percent of the observed total volume were considered.

The second approach is to let each company's volume of business vary according to a given distribution around it's observed volume to see what HHI values resulted. This approach may be more meaningful in that it is more intuitive. The same total volume restriction was also employed here.

Although there is literature to consider to determine the number of replications needed, here it was very simple to experiment with numbers of replications differing by one or more orders of magnitude to see what practical changes occur in the results of interest. (Note that this is similar to what was done in Knaub (1985b), page 457, although a modification of the procedure found in Knaub (1985a), such as that illustrated in Knaub (1986), could be used.)

Conclusions:

From the table on the next page, it may be concluded that one should be wary of

1) forecasts of change based upon trend analyses supported only by a change in the HHI of five percent or smaller, and
2) forecasts of constancy based upon trend analyses where the HHI has changed by fifteen percent or more.

Addendum:

Suppose a national market were to be considered by State. Small changes in HHI may indicate a trend if enough of the State markets had HHI changes in the same direction. Confidence intervals or hypothesis testing, considering both types of error, could be used to determine whether the trend was substantial.

Tabular Examples of Simulation Results:

	A	B	C	D	E	F
Att. HHI	4021	2193	605	434	374	324
NC	52	81	106	272	392	87
B						
U						
MEAN HHI	3150	-	599	-	359	319
MED. HHI	2887	-	587	-	360	314
HHI CV	44	-	17	-	20	14
R						
MEAN HHI	3968	-	601	-	368	320
MED. HHI	4018	-	597	-	367	318
HHI CV	11	-	13	-	18	14
IR COMPANY CVs=5						
U						
MEAN HHI	4021	2193	607	435	375	325
MED. HHI	4021	2195	607	435	375	324
HHI CV	3.3	4.6	1.7	2.4	2.7	1.9
R						
MEAN HHI	4021	2193	607	435	375	325
MED. HHI	4020	2194	607	435	375	324
HHI CV	3.2	4.5	1.7	2.4	2.7	1.9
IR COMPANY CVs=10						
U						
MEAN HHI	4020	2194	611	438	377	327
MED. HHI	4022	2199	611	438	377	326
HHI CV	6.6	9.2	3.5	4.9	5.4	3.8
R						
MEAN HHI	4027	2195	611	438	377	327
MED HHI	4019	2192	611	438	377	326
HHI CV	5.8	7.7	3.4	4.8	5.4	3.8

"A," "B," "C," "D," AND "E" represent retail motor gasoline in five States
"F" represents residential distillate in one State
"-" indicates data not collected
"R" represents total volume restricted to + or - 5% of the observed total volume
"U" represents "unrestricted" volume
"B" denotes a "bootstrap-like" simulation
"IR Company CVs=x" denotes the second simulation approach where individual replacement of each company's sales volume occurs using a normal distribution with mean equal to the observed volume, and CV=x
"Att. HHI" is the attained HHI
"NC" is the number of companies in the frame
"MEAN HHI" is the mean of the simulated HHIs
"MED. HHI" is the median of the simulated HHIs
"HHI CV" is the CV of the simulated HHIs

References:

Knaub, J. R., Jr. (1985a), "Nonparametric Median Estimation (With Application to Number of Simulation Replications Needed)," in Proceedings of the Thirtieth Conference on the Design of Experiments in Army Research, Development and Testing (Vol. 2), Research Triangle Park, NC: U.S. Army Research Office.

------(1985b), "On the Lehmann Power Analysis for the Wilcoxon Rank Sum Test," in Proceedings of the Thirtieth Conference on the Design of Experiments in Army Research, Development and Testing (Vol.2), Research Triangle Park, NC: U.S. Army Research Office, pp. 455-478.

------(1986), "Study of Supplementary Sampling of Food Stamp Quality Control Data," Proceedings of the National Association for Welfare Research and Statistics.

U.S. Department of Justice (1982), "Merger Guidelines," Issued: June 14, 1982.

Encoding and Processing of Chinese Language--A Statistical Structural Approach

Chaiho C. Wang*
U. S. Department of Justice and The George Washington University

Introduction

The performance of the encoding and transmission of language information may be measured by the following criteria:

1. Preserving culture identity,
2. Maximizing processing speed,
3. Maximizing transmission accuracy, and minimizing ambiguity,
4. Minimizing human effort,
5. Minimizing storage requirement.

On the one hand, the Chinese language has had the time to grow deep cultural roots unsurpassed by any other in existence. On the other hand, it had been developed by, and served the relatively few educated scholars. For centuries, the practicality of teaching it to the masses has not been a priority. Today, when rapid data transmission has become so important in life, the structure of the Chinese language provides information processors a great challenge.

This paper proposes mathematical models in which a balanced approach among the five criteria is considered. Based on the statistical structure of the Chinese language, this procedure incorporates a user friendly input coding scheme with a low redundancy internal coding method for compressed storage. Both the graphical and pinyin input options are considered, and special attention is paid to reduce human effort at the data entry stage. With the new technology of tomorrow in mind, the goal of efficiently computerizing Chinese language may be within reach.

Basic Encoding Methods

Let $L = \langle V,B \rangle$ be a language where V is a vocabulary, and B is the set of basic symbols (messages). The language may be composed of elements of V, which are strings of basic symbols of B. Let C be a set of codes which may be used to represent elements of B, and D be a set of internal codes which represent the elements of C within a computer. The set D may be machine dependent, and need not be of concern to the user. In a Shannon [10] like theory, the average message length (entropy) is defined by

$$(1) \qquad E = \sum_{i}^{m} p_i \log p_i$$

where p_i is the probability of occurrence of the ith message, and m is the number of elements in B. Since binary coding digits are customarily used, the symbol "log" stands for the logarithm of base 2, and the unit length is called bit per message, or bit per character (BPC). If a code c of the set C represents more than one, say t, messages from B, then c is said to be of multiplicity t.

As an example, the basic elements B of the English language may be taken as the set of 26 characters (plus a few special characters necessary to form a grammar), while V may be a set of words, or a set of fixed-length character strings. The usage of these characters, such as in frequency distributions, redundancy, and storage requirements, can be readily studied (see [1, 6, 9, 11, 12, 14, 15]). Hereafter, we let "English" stand in general for any alphabetized Western language.

The human effort of encoding English text normally involves a one step phonetic process: whether the typist is listening to a dictated message, or reading from a document, the sound of a word, say "teacher", is translated into the correct spelling, t-e-a-c-h-e-r, which is then entered, letter by letter, on a keyboard.

In a straight forward coding of English langauge, assuming that all characters occur with equal frequency, E = log 26, which is about 5 bits per character. On the one hand, a standard storage cell is either 6-bits or 8-bits in size for a standard main frame computer, which provides for use of both upper and lower case letters, and manipulation of information beyond English text. On the other hand, the use of compressed storage algorithms (see [2, 3, 6, 9, 11, 12]), which use numerical coding, language elements coding, or probabilistic coding, allows the reduction of E to below five bits. For example, the Huffman [6] minimum redundancy variable-length codes take advantage of the frequency distribution of the occurrences of elements of B, reducing E to 4.2. An alternative method, which utilizes fixed-length codes while splitting B into groups (see [12]), can achieve a similar result. Here the following formula is used to compute entropy:

$$(2) \qquad E = \sum_{i}^{G} p_i (\log p_i + \log G)$$

where p_i is the proportion of usage for the ith group ($\sum p_i = 1$). Here within each of the G groups a fixed length code is used, and a log G-bit flag is used to identify the groups.

For modeling Chinese, as a first attempt we consider a straight coding method, in which B is the set of all Chinese characters, and C contains enough codes to represent elements of B one-to-one. Depending on the application, B can have several thousand to tens of thousands of elements. Although a keyboard containing the entire set B can be made available, the time required to search the keys make data entry prohibitively difficult.

In order to locate a given character, the set B is structured into groups according to shapes of "radicals" (部首), and the elements within a group are sorted according the number of strokes in a radical or a character. This is the standard dictionary lookup method, which was developed years ago without automation in mind. Although a computer method can be devised to imitate the dictionary lookup, it is not practical. This model can be considered as a strict cultural approach, where the meanings and shapes of the radicals and characters are recognized.

The greater the size of B, the more cumbersome the data entry process, but a large entropy is not necessarily a consequence. Comparing the storage requirements of a document written in Chinese with its English translation shows that, generally, less storage space is needed for the Chinese version than for its English translation. Assume that a six bit unit is used to encode an English alphabet, allowing space for both upper and lower case letters, and a 14 bit unit is used to encode a Chinese character. Simple experiment indicates that for coding newspaper information, the ratio of storage space for the Chinese text to its English translation is two to three. For coding classical Chinese (wenyen 文言), the ratio is one to four. (Source documents for the experiment are: 1. China Reconstructs Magazine, published both in Chinese and in English, in Beijing. 2. Yen, L. (1976), A reconstructed Lao-Tze with English translation, Cheng Wen Publishing Co., Taipei.)

To facilitate dictionary lookup, in a second attempt, numerical codes were developed to represent the strokes, radicals, and the shape of a character. This approach goes back several decades. The four corner coding method is an good example. Several such methods have recently been developed on a computer (see, for example, [4, 5]). In this case, the set C contains "structural" codes, representing the composition of strokes in a character.

During data entry, a typist observes a character, codes it according a set of rules, and enters the numerical code on the keyboard. The method requires a considerable amount of human effort--the coder must memorize the codes for the radicals and the encoding rules. In addition, large work space and storage space may be required. For example, with the three-corner method [5], three two-digit fields are required; that is, 999,999 positions are required to represent, say 10,000 characters. Moreover, unless the numerical code is unique, an additional auxiliary code must be used to eliminate multiplicity.

If an one-to-one correspondence is established between the set of codes C, and B, then the coding is unambiguous, and decoding becomes possible. If several elements of B share a common code, special measures must be taken to assure accuracy and decodibility of messages. We shall call the former an one-to-one method, the latter a one-to-many method. Neither the four-corner, nor the three corner method is one-to-one. To make up for the deficiency, an auxiliary code is needed to represent all messages that belong to the same code.

As a third attempt, we rely on the pinyin method -- the phonic approach common to the majority of Western languages. Here B is the Latin alphabet. This gives rise to a two step data entry process: the typist observes the characters, say " ", translates it into is phonetic representation, "shi", then enters it into the keyboard. Since "shi" also represents many other words (one-to-many), a secondary code is required to single out " ".

As will be discussed in the next section, the pinyin method has several attractive features. Comparing pinyin usage of the alphabet with English there are several dissimilarities: (1) In English, the alphabet symbols are codes as well as language elements. What you see is what you code. In pinyin, the pinyin symbols are codes, not language elements. A coder must first translate a character into its pinyin representation, then enter the code in the keyboard. (2) Non-standardization of pronunciation of characters give rise to inaccuracy problems. (3) There may be many characters with the same pronunciation, therefore, making decoding difficult. (4) Each character may have four or five tones, which will contribute to inaccuracy and identification problems.

Statistical structure

In a straight coding of English by singleton characters, E = log 26. After the letters are arranged according to their frequency of occurrence, in

descending order, and the cumulative frequency distribution is tabulated, they can be split into two or more groups. For example, among the 26 letters, the eight most commonly used letters account for 60 percent of the usage. We have

$$E = .6(\log 8 + 1) + .4(\log 18 + 1) = 4.46$$

which is greater than the Huffman entropy, but less than log 26 = 4.7.

For a single-character splitting of Chinese characters, let m be the number of characters, and assume that the first x most frequently used characters account for p percent of the usage. Then a split between the two groups of x characters and the remaining m - x characters yields the following entropy

$$E = p(\log x + 1) + (1 - p)(\log(m - x) + 1)$$

Assuming that m is sufficiently larger than x, the splitting would keep E unchanged if

$$p(\log x + 1) + (1 - p)(\log m + 1) \approx \log m$$

It follows that

$$p = 1/(\log m - \log x).$$

If m =8000, and x = 500, then p = 1/4. This says that if 500 common characters account for at least 25 percent of the usage, then numerically a splitting will decrease entropy.

In an early Book by Zipf [15], frequency distribution of the usage of Chinese characters is tabulated (see Table 1). This table is produced based on a sample of 20000 syllables of speech in Beijing dialect, which are arranged according to their frequencies of occurrence. Reading Table 1 from the left, the first column gives the number of occurrences, the second column presents the number of words assigned to a given frequency of occurrence, and the other columns indicate the number of words in each frequency group having one or more syllables.

Applying Huffman minimum-redundancy coding method to Table 1 data, E can be computed as 9.654 bits per word. Since there are 13,118 words among 20,000 syllables, the Huffman entropy per syllable would be lower than 9.654. In Zipf's work, the relationship between a syllable and a character was not clearly defined. Therefore, the actual (character) entropy cannot be deduced. From this table, however, we can determine that, approximately five percent of the syllables account for fifty percent of the usage, ten percent of the syllables account for sixty percent of the usage, and forty percent of the syllables account for eighty five percent of the usage. For these three profiles, using formula (2), we find E equal to 10.6, 10.4, and 10.8, respectively.

Number of Occurrences	Number of Words	Number of Words with their Syllables					
		Number of Syllables					
		1	2	3	4	5	6
1	2046	315	1571	144	14	1	1
2	404	110	358	23	3		
3	216	59	147	9	1		
4	100	24	73	3			
5	99	39	58	2			
6	66	24	41	1			
7	41	16	25				
8	25	10	14	1			
9	30	13	15	1	1		
10	20	13	7				
11	25	14	11				
12	22	15	7				
13	10	6	4				
14	14	7	7				
15	13	5	8				
16	10	4	5		1		
17	10	6	4				
18	6	2	4				
19	5	4	1				
20	5	5					
21	4	3	1				
22	2	2					
23	5	4		1			
26	3	2	1				
28	4	3	1				
29	4	1	3				
30	6	4	2				
32	6	4	2				
33	2	1	1				
34	1	1					
35	1	1					
36	1	1					
37	1		1				
38	1		1				
41	4	4					
43	2		2				
44	2	1	1				
45	3	1	2				
46	1	1					
47	2	2					
50	1		1				
52	1	1					
55	2	2					
57	1	1					
58	1	1					
60	1	1					
66	2	1	1				
68	1	1					
72	1	1					
73	1	1					
75	1	1					
78	1	1					
81	1	1					
83	1	1					
101	2	2					
102-905	12	12					
13,248	3,332						

Table 1. Chinese of Beijing
(Adopted from G. Zipf: The Psyco-Biology of Language)

In today's standard classification [8], there are two classes: class one of 3,755 common characters, and class two of 3,008 uncommon characters. If the 6,763 characters are treated equally, the entropy is 12.7. If, similar to Zipf's findings, five percent, or 338 characters account for 50 percent the usage, then

E = (log 338 + log 6425)/2 + 1 = 11.5, which represents more than a fifty percent storage reduction over the direct method. If the vocabulary is to be expanded to cover uncommon Chinese characters, in the range of, say, 30,000 to 50,000 characters, the splitting method will even be more useful.

Frequency tables such as Table 1 can be created based on the domain of information to be processed. For example, a vocabulary in chemistry would be different from that for composing children's books. Clearly, storage compression can be made more effective if the correct vocabulary is defined.

Next, we consider the structure of the graphical Chinese symbols. According to the standard classification, there are 167 radicals. Conveniently, all characters can be divided into 167 groups using the leading radical as an index. For the moment, let us call the leading radical and the remaining parts of the character the "head" and the "body" of a character, respectively. For the purpose of linguistic studies, radicals are being "isolated" according to their meanings and historical background. The more accurate the classification, the larger becomes the collection. For reducing human effort in data processing, however, the set of codes for radicals should be reduced. Because the Chinese vocabulary has been carefully developed and refined over the centuries, two disjoint radical groups can be merged with little risk of ambiguity. For example, the groups with leading radicals "wen" (文), and "fanwen" (攵), have 12, and 32 characters, respectively; but there is no overlapping among the bodies of the total 44 characters. That is, if we use the same leading code for wen and fanwen, a one-to-one correspondence between the 44 characters and their designated codes is preserved; and a computer will have no trouble distinguishing between them.

Suppose we throw in another group with a leading radical similar in shape to wen, say the "tongzhitou" (夂), we add another ten characters to the list. At this point, we finally encounter one overlap of a character body: (口). When this occurs, we can simply create a secondary code to differentiate between (各) and (各).

If we can regroup and reduce the set of codes for radicals to, say, a total of 64, a pleasant keyboard containing these radicals can be developed, and a data entry method based on graphical coding becomes feasible.

The pinyin method adopts the 26 Latin letters as the basic symbols. At a first glance, the basic analysis for other Western languages will be applicable to pinyin. A closer look reveals two distinctions. First, among the 26 letters, 26^2 diagrams, 26^3 trigrams, ..., only 407 pinyin units, from one to six letters were actually used! With a few exceptions, a pinyin unit is formed by adjoining a leading code and a terminal code (similar to the use of consonant and vowels in English). Since there are 23 leading elements and 33 terminal elements, the maximum possible number of pinyin units is 23x33 = 759.

Second, each pinyin unit may represent between one and 115 distinct characters. If an auxiliary code is used to handle the multiple representation we need 407x115=46805 codes (E=15.5). In order to reduce the code size, the 407 pinyin units may be arranged according to the multiplicity of each unit, and a variable size auxiliary code is used to differentiate the elements within a group.

A mathematical model

Based on the statistical properties of the Chinese language, we propose a model for encoding, storage, and transmission of Chinese language data. This model includes (1) an internal "minimal" redundancy code that stores information efficiently. (Not necessarily an absolute *minimum*, but rather a *minimum* with respect to a particular application). (2) a user friendly input method which "minimizes" the human effort. The input method can be based on either the graphic or the pinyin approach. Finally, (3) an automated dictionary "lookup" method which links these two devices together.

The Internal Code.

First, a vocabulary of characters, such as the standard collection of the 6763 characters, is defined. Next, an experiment is conducted to determine the frequencies of occurrence of the characters. Based on the statistics, a user preferred data compression method, which facilitates the sorting, merging and other data processing tasks, is developed. Common phrases can sometimes be coded as single messages to further reduce storage. If the 6763 characters were treated equally, the entropy is 12.7. This may serve as a guide for judging the performance of the new coding scheme. As we have seen earlier, the Zipf data, in which five percent, or 338 messages account for 50 percent the usage, yields an entropy of 11.5.

The Input Mode

The input can be either in a graphical coding mode, a pinyin mode, or a mixed mode, such as the method adopted by the

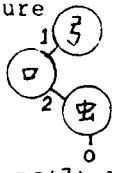
Chinese National Bureau of Standards for classifying the Chinese characters (see [8]).

The Graphical Method. First, the number of radical groups must be reduced, as described in the previous section. A collection of no more than 64 basic radicals would be suitable for quick recognition on a keyboard. Next, as the encoding begins, a given character is decomposed into a number of basic radicals, which are linked together by following the natural sequence of drawing the strokes of the character. The code is a linked list of radical code (RC) and direction code (DC).

RC_1	DC_1	RC_2	DC_2	...	RC_n	0

The direction code can be simply defined as a 2-bit pointer, say, "1" for a top-down movement, "2" for a left-to-right movement, and "0" as a nil pointer indicating the end of the character code.

For example, the character (㠯) will have the structure



and the code RC(㠯)-1-RC(口)-2-RC(虫)-0

Here the code consists of three 6-bit radical codes and three 2-bit direction codes, for a total 24 bits. Although some of the characters can have a quite long code, the average length per character will be around 22 bits. After the complete set of these variable length codes are sorted, a look-up table for decoding can be set up.

If fixed length codes are desired, a numerical code similar to the four corner, or three corner method can be developed. A typical code, based on regrouping of radicals, will look like

PC	RC_1	RC_2	MC

where PC is a shape-code describing how the character is partitioned into radicals, RC_1 and RC_2 are the two radical codes, and MC is an auxiliary multiplicity correcting code pinpointing the given character within the radical group. The ordinary three corner method requires a 20-bit (fixed length) code. With an auxiliary code, the total length would be raised to about 25 bits. The basic four corner code is only 14 to 15 bits, but its auxiliary code would be quite large.

The phonic method. The popularity of the pinyin method depends on the extent the pronunciations of speaking Chinese is unified. For users who have acquired the pronunciation skills, this method is very promising. Standard pinyin keyboards have been developed and improved (see [7, 13]). A typical pinyin code would have the form

LC	TC	MC

where LC and TC denote the leading and terminal pinyin codes, respectively. MC is the auxiliary code required to eliminate multiplicity.

In the worst case, a full pinyin code would have 23x34x115 = 89,930 messages, which translates to 16.5 BPC. Since there are only some 407 active pinyin units, the total number of messages is reduced to 403X115 = 46,345, or 15.5 BPC. But the auxiliary code can be reduced too. Since the degree of multiplicity varies among pinyin units, the auxiliary code can be made into variable length. When the basic pinyin unit codes (LC-TC) are sorted according to the values of the auxiliary code, the units with a large auxiliary lookup table are separated from the rest. Only when these units are called for, need one allocate maximum space for processing the auxiliary lookup table.

If a fixed length auxiliary code is desired, the pinyin unit which has a large multiplicity can be divided into two or more records in the following form:

LC	TC	AC_1	LC	TC	AC_2	...

Since among the 407 pinyin units, only 50 have a multiplicity 32 or higher, a 5-bit auxiliary code would be sufficient. The unit with the greatest multiplicity (115), will be divided into four records.

To take full advantage of the fact that there are only 407 active pinyin codes, a voice activated procedure can be developed for data input. After the computer is programed to recognize these 407 sound patterns, a coder, while input a data set, may sound out the characters one by one. Each time, the computer would identify the pinyin unit, prompt the coder with a screen full of characters belonging to the pinyin unit. To maximize efficiency, for each pinyin (group) code, the corresponding auxiliary set of characters is presorted according to their frequency of usages. After the coder points to the designated character, it is automatically coded.

Summary

At first glance, the Chinese langua(is too complicated for automated processing. A closer look at its statistical structure, however, reveals many "built in" features suitable for efficient encoding and processing. Much experimentation will be needed to shade light on the statistical structure of the usage of Chinese characters. The encoding procedures proposed in this paper may sound too good to be true; but the technology is available, the theory is simple, and the potential is promising, therefore, further research in this direction is warranted.

* The views expressed in this paper do not necessarily reflect those of the Department of Justice.

References

1. Bourne, C., and Ford, D. (1961), A study of the statistics of letters in English words, Information and Control, 4, 48-67.
2. Hagaman, W., Linden, D., Long, H., and Weber, J. (1972) Encoding verbal information as unique numbers, IBM Systems Journal, 11, 4, 278-315.
3. Hahn, B. (1974), A new technique for compression and storage of data, Communications of the ACM, 17, 8, 434-436.
4. Huang, H. (1986), The Huang Chinese encoding method, Private communication.
5. Huang, J. (1985), The input and output of Chinese and Japanese characters, Computer Magazine, 18, 1, 18-23.
6. Huffman, D. (1952), A method for the construction of minimum-redundancy codes, Proceeding of the I.R.E., 40, 9, 1098-1101.
7. Jian, S. (1985), A pinyin keyboard for inputting Chinese characters, Computer Magazine, 18, 1, 60-63.
8. The People's Republic of China National Bureau of Standards (1981), 国家标准信息交换汉字编码字符集基本集.
9. Schwartz, E. (1967), A language element for compression coding, Information and Control, 10, 315-333.
10. Shannon, C. (1948), A mathematical theory of communications, The Bell System Technical Journal, 27, 379-423.
11. Wagner, R. (1973), Common phrases and minimum-space text storage, Communications of the ACM, 16, 3, 148-152.
12. Wang, C. (1980), A probabilistic approach to storage compression of large natural language data bases, Proceeding of the IEEE computer Society's Fourth International Computer Software & Applications Conference, 552-556.
13. Wei, C. (1985), Evaluation of Chinese character keyboards, Computer Magazine, 18, 1, 54-59.
14. Zipf, G. (1935), The Psycho-biology of language, a introduction to dynamic philology, Houghton Mifflin, Boston.
15. Zipf, G. (1949), Human behavior and the principle of least effort, a introduction to human ecology, Addison-Wesley, Cambridge, Mass.

XVIII. BIOSTATISTICAL METHODS

AN ALGORITHM TO IDENTIFY CHANGES IN HORMONE PATTERNS

Morton B. Brown[1], Fred J. Karsch[2] and Benoit Malpaux[2]
The University of Michigan, Ann Arbor, MI

ABSTRACT

Many hormones are secreted into the blood in a pulsatile manner: i.e., in high concentrations at 'random' times. To study hormone production, investigators assay its level in the blood at regularly spaced intervals. The statistical problem is to differentiate between changes in the level of the hormone and observations influenced by a 'random' pulse ('noise'). Two algorithms are described: One uses regression-like statistics computed after deleting the most 'extreme' observation combined with a moving variable-length window to identify rises and declines in hormone level. The deletion of the most 'extreme' observation and the use of a variable-length window facilitates the exclusion of 'noisy' values from the determination of the stage of the hormone. The second algorithm uses a least-squares criterion to cluster adjacent points after the elimination of 'noisy' values. A test statistic for termination of the clustering is described.

Keywords: cluster analysis, pattern analysis, regression, biological rhythms

INTRODUCTION

It is believed that hormones regulate many different time-dependent processes such as fertility cycles (Bittman et al 1983a,b; Farner and Follett, 1979; Robinson and Follett, 1982). Some rhythms are annual, others are monthly, and yet others are daily. Many investigators are studying the manner by which this time-dependence is regulated and how the time-dependence can be disrupted (Nett and Niswender, 1982; Robinson and Karsch, 1984). For example, many annual rhythms are synchronized by the number of hours of daylight: maintaining animals in a light-controlled environment and modifying the length of the light period is used to study the effect of disrupting this stimulus. Similarly, many daily rhythms are affected by the time of day and are synchronized by the light/dark cycle.

Although it is anticipated that hormone production is regulated in some manner (by responding to a stimulus which may be another hormone), it is not unusual for some amount of the hormone to be present in the blood stream even when production of the hormone is in a reduced state. It is now recognized that some hormones are released in a pulsatile manner from the gland in which they are produced. That is, a high concentration of hormone (a pulse) is released into the blood over a relatively short interval. The hormone is then extracted from the blood as it passes through an organ, such as the liver, or mixes rapidly with the blood during the next few passes through the circulatory system.

Experiments to understand the time pattern of hormone production involve repeated sampling of the blood; the samples are then assayed for the amount of hormone. The number of samples is limited by the cost or by the amount of blood that can be removed without causing damage to the subject (human or animal). Usually, continuous sampling is not feasible. The number of samples is determined by the length of time over which samples are required: the greater the length of time, the greater the spacing between samples. When studying hormone levels during an 8 hour period, it may be possible to take 2 to 3 samples per hour; if the period is only 4 hours, 5 to 6 samples per hour may be taken. In contrast, when studying an annual rhythm, only two to three samples per week may be possible.

Often the samples are assayed by radioimmunoassay methods to estimate the amount of hormone (Niswender et al 1969). For the purposes of this paper we will assume that the method of estimating the hormone has been standardized. Since high values usually have larger variability than small values, a logarithmic transformation is often applied before any analysis. This transformation will be performed before all analyses described in this paper.

The statistical problem is to identify when the level of hormone is elevated as compared to when it is at baseline. If the pattern is expected to consist of cycles of baseline and plateau; each cycle can be characterized by four states: baseline, a rise, a plateau and a decline, followed again by baseline, etc. In order to compare the effects of different interventions, it is desirable to have an objective method to identify these four states and the times at which changes between the states occur.

When this pattern is expected to repeat itself at regular intervals, approaches to the estimation of the frequency of pulses are spectral analysis (Koopmans, 1974) or the fitting of ARIMA models (Box and Jenkins, 1976). Two problems with these approaches are that: (1) because of limitations on the amount of blood drawn, often only one cycle is observed in any subject, and (2) many experiments are designed to disrupt the rhythm so that the series will not be stationary.

Any sample taken during, or shortly after, the release of a bolus (a concentrated pulse) will show very high levels of hormone. These boluses are released at randomly spaced times, more frequently when production of the hormone is rapid and less frequently when the production of the hormone is at a nadir. However, if the blood sample is taken near the time of a pulse (whether at the peak or nadir of the cycle), a high level of hormone will be found in the blood. At the nadir of the cycle, it is less likely that the values of several successive samples will all be elevated.

Therefore, a statistical model for the hormone would include the four phases (baseline, rise, plateau, and decline). The error term would be composed of two parts, one conventional (due to random biological and technical variation) and a second probabilistic to reflect the possibility

of sampling at or near the time of a bolus even when the overall level of hormone may be low.

In this paper we will first describe an empirical algorithm to identify the four phases of the cycle. The core of the algorithm is to identify the four phases by using regression statistics that are computed about each time point using limited sets of continguous data values. A drawback to this algorithm is the difficulty in specifying a statistical criterion or test for the presence of a cycle.

Therefore, a second algorithm is also described which clusters contiguous points using a least squares algorithm. A conservative t-like statistic is proposed as a criterion to terminate the clustering process.

Although more work is needed on the development of these algorithms, the use of an algorithm (even one that is not optimal) provides results that can be compared across interventions and is preferable to the subjective evaluations of cycles and phases that are currently used.

ALGORITHM 1: USING REGRESSION STATISTICS

The model assumed for the data is that there are four phases:

baseline	$y_t = c + e_t$	
rise	$y_t = a + bt + e_t$	$b > 0$
plateau	$y_t = c + e_t$	
decline	$y_t = a - bt + e_t$	$b > 0$

where a, b and c are constants that differ between phases and between cycles and b is strictly positive. Each model holds only for a single phase which is represented by a restricted interval in time (t); amplitudes of the baselines and of the plateaus, and the slopes of the rises and declines will differ at different cycles (time intervals). That is, no regularity of the signal is assumed.

The error e_t is composed of two parts:

$$e_t = E_t + h(t-t_0)$$

where E_t is an error term with mean zero and constant variance and $h(t-t_0)$ is a function that represents the height of the signal due to a bolus released at time t_0 but assayed at time t. The function $h(t-t_0)$ is positive, but may be zero when the sample is taken sufficiently far (in time) from the last bolus so that the net effect of the bolus is negligible (part of the baseline or plateau). Note that the effect of h on the error term is asymmetric.

The first attempt at developing the algorithm was to fit line segments to the data using a fixed window (i.e., using the same number of contiguous points each time) and centering the window at each time point in the data sequence. Not surprisingly, single high values (caused by sampling near a bolus) produced estimates of slopes that were positive when approaching the point and negative after the point. That is, values caused by a bolus were highly influential in determining the estimates of the coefficients and therefore in determining the phase. Hence, the following approach was used.

Let t represent the center of a window (set of contiguous points) and g the number of data points in the window to each side of t. That is, the data points in the window are

$$y_{t-g}, y_{t-g+1}, \ldots, y_{t-1}, y_t, y_{t+1}, \ldots, y_{t+g-1}, y_{t+g}.$$

For each t and a set of g's ($g_1 \le g \le g_2$), estimate the regression statistics:

the mean of y:	$\bar{y}$
the variance of y:	s_{yy}
the covariance of y and t:	s_{yt}
the variance of t:	s_{tt}
the slope of y on t:	$b = s_{yt}/s_{tt}$
the correlation of y and t:	$r = s_{yt}/(s_{yy}s_{tt})^{\frac{1}{2}}$

The above statistics were then corrected to eliminate the most discrepant observation in the window. (Initially, this criterion was used to eliminate the maximal value only but examination of the signal indicated that at the time that pulsing slowed, single isolated low values may occur in the signal. Therefore, the criterion was modified to be symmetric.) This adjustment is applied to all windows in a uniform manner so that statistics across windows are comparable.

There is no unique width of window that is optimal. A single width is not desirable, since it reduces the flexibility of the algorithm to smooth across data values and fulfil a criterion for identifying a baseline or plateau. The use of a variable length window (i.e., windows of length 7, 9 and 11 were used simultaneously) allows greater smoothing across data values and therefore less very short cycles are identified. (Very short cycles are not believed to be of physiological interest.)

We have chosen to use empirical critical values to determine the phases, where the values are estimated by percentiles of the empirical distribution. For each of the above statistics, the median and the quartiles are estimates. Two exceptions should be noted. Since the baseline in some hormones may be below the threshhold of the assay and therefore s_{yy} may be zero for many evaluations, the quartiles for s_{yy} are computed from nonthreshhold (minimum) values. For slope, the upper and lower quartiles correspond to the medians of the positive and negative slopes respectively.

The criteria for determination of *possible* phase at a time point are:

baseline: $(\min r^2 < Q_{25}$ OR $\min s_{yy} <$ median) AND $y < Q_{75}$

rise: $\max r_+^2 > Q_{75}$ OR $\max \text{slope}_+ > Q_{75}$ OR $(s_{yy} > Q_{75}$ AND $\max \text{slope}_+ > |\max \text{slope}_-|)$

plateau: $(\min r^2 < Q_{25}$ OR $\min s_{yy} <$ median) AND $y > Q_{25}$

decline: $\max r_-^2 > Q_{75}$ OR $\max \text{slope}_- < Q_{25}$ OR $(s_{yy} > Q_{75}$ AND $\max \text{slope}_+ < |\max \text{slope}_-|$

where min and max are the minimum and maximum, respectively, of the statistic over all the win-

dows for which the statistic is computed. Q_{25} is the lower quartile and Q_{75} is the upper quartile for all values of the statistic (as computed with the exceptions described above). The + or − sign used as a subscript indicates that those statistics that are computed only when the slope is positive or negative, respectively; i.e., those appropriate to identify a rise or a decline, respectively. AND and OR are the logical operators: both sides of an AND must be fulfilled for the condition to be accepted, but only one (or both) side(s) of an OR needs to be fulfilled.

Based on the above definitions of phases, many time points can be assigned to more than one phase. Therefore, the final phase for each time point is chosen using the philosophy that the method of allocation should favor a point being set to a baseline or plateau when the set of neighboring points have low variablility (or low correlation with time). Only when there is sufficient variability within a contiguous set of time points should the phase be set to a rise or decline (depending on the sign of the coefficient). This is implemented in the following manner: *(Lower case letters indicate that the criteria for the phase are fulfilled and capital letters indicate that the phase has been assigned to the time point.)*

Pass 1:

IF (baseline AND NOT plateau) THEN BASELINE

IF (plateau AND NOT baseline) THEN PLATEAU

IF (ONLY rise) THEN RISE

IF (ONLY decline) THEN DECLINE

Pass 2:

In this and the next passes, contiguous points that are as yet unallocated to phases but have the same set of possible phases are treated as a single time point in terms of determining the preceding and following phases.

IF (baseline OR plateau AND:

continguous to point set to BASELINE)
THEN BASELINE

continguous to point set to PLATEAU)
THEN PLATEAU

follows a point set to DECLINE)
THEN BASELINE

follows a point set to RISE)
THEN PLATEAU

precedes a point set to RISE)
THEN BASELINE

precedes a point set to DECLINE)
THEN PLATEAU

Pass 3:

All unallocated points are set to a phase that is most consistent with the phases of the neighboring time points.

All as yet unallocated time points that have a permissible phase equal to a contiguous point which has already been assigned its phase are set to the phase of the neighboring point. Otherwise, if a permissible phase is consistent with a phase that should follow the phase of the preceding point or to a phase that should precede that of the following point (if valid), then that phase is selected for this time point. Those time points which do not fulfill the requirements of any of the phases are set equal to a neighboring phase.

Discussion of the Regression Approach

Several aspects of this algorithm need further explanation:

When using a short window, a high correlation may occur even when only one point differs from all the other points; e.g., the correlation of time to a sequence of k values that are constant except for one endpoint which is unequal to the other values is $(3/(k+1))^{\frac{1}{2}}$; i.e., when k is 9, the correlation is 0.55. Therefore, high correlations can occur in the presence of low variation. By assigning all cases to the baseline or plateau when there is low variation, high correlations by themselves do not cause these times to be classified into a rise or decline.

We have already described the problem of high values near the time of release of a bolus. Since it is not possible to predict when these will occur, the algorithm must be relatively insensitive to large spikes. The blood will usually complete several full circulations through the body between two successive samples; therefore, the high level of hormone released by a bolus should primarily affect the value of a single observation. (Under very rapid sampling it is possible that more than one consecutive sample will elevated by a single bolus.) The elimination of the maximal value in each window reduces the effect of a possible spike. We do not suggest testing for a spike prior to eliminating a point since the distribution of the elevated value is a continuum and the observed value depends on the timing of the sample relative to the actual release of the bolus.

Empirical values are used to determine the assignment of time points to phases. The evaluations of the statistics reuse the same data many times: both for different windows centered at the same time point and also for windows centered at continguous or nearby time points. Therefore, the set of statistics generated in the first phase of the algorithm are highly correlated. For this reason it would be difficult to develop exact tests of significance. For example, for each window we also computed the F-statistic that tests whether the slope (or correlation) is zero. Since definite cycles exist in the data that we have analyzed, the F-statistics corresponding to the quartiles are highly significant.

The criteria for rise and decline use quartiles. The criteria for baseline and plateau use the median. This is again a definite bias in the allocation scheme to set time points to a 'flat' phase, rather than a 'changing' phase. Our experience has shown that setting more restrictive criteria for the baseline and plateau causes many short cycles to be identified within a plateau or a baseline. These new cycles are very short and do not correspond to our understanding of the underlying physiology of the rhythm under study.

The algorithm is designed to choose among the possible phases and to select that phase that enhances the cycling. Therefore, the clean appearance of the result is not a proof of the cycling. As indicated above, we assume that other more standard methods have been used to

test for changes in the height of the signal, which corresponds to testing for the presence of at least two different levels of activity. This algorithm is intended to identify the location of the cycles.

ALGORITHM 2: CLUSTERING NEIGHBORS

Using the same model as above, it is possible to view the time sequence of data points as divided into clusters of time points with similar values: a cluster with a low mean value is a baseline and one with a high mean is a plateau. Between these two clusters may be one or more clusters with intermediate values corresponding to a rise of decline.

The algorithm for clustering adjacent points consists of the following steps:

Step 1:

Compute the minimum, maximum, range and standard deviation (SD) of the sequence. Identify as an outlier any point that differs by one-half the range from the mean of its four neighbors (two time points on either side) and is the minimum or maximum of these five points. Outliers are then eliminated from calculations in the remainder of the algorithm.

Step 2:

Set each time point to be a cluster. Combine adjoining time points if they differ by less than 1% of the range. (This eliminates the need to combine similar observations by repetitive steps in the algorithm described below.)

Step 3:

Compute the distances between adjoining clusters and clusters separated by a single intermediate cluster where the measure of distance is a t-like statistic defined by:

Let

$$t = d / SD$$

where

$$d^2 = n_1 n_2 (\bar{x}_1 - \bar{x}_2)^2 / (n_1 + n_2)$$

and $\bar{x}_1, \bar{x}_2$ are the means of the two clusters and n_1, n_2 are their sample sizes. Then t is distributed as a t-statistic when there are no clusters in the series and will be bounded above by the distribution of a t-statistic when there are clusters (since the within cluster pooled standard deviation must be less or equal to the SD from the entire series).

Let t_2 represent the measure of distance between two adjacent clusters and t_3 that between two clusters that are separated by a single cluster. Min t_2 and min t_3 will represent the minimum values of these statistics across all the clusters.

Step 4:

Combine the two adjoining clusters with min t_2 unless: (1) two clusters with similar mean values are separated by a single point ($t_3 < min\ t_2$), or (2) the cluster formed by combining the two clusters with min t_2 could be redivided into two clusters with a smaller within cluster sum of squares, or (3) the two clusters are part of a sequence of three consecutive clusters such that the distance between the two outermost clusters (t_3) is larger than a criterion (discussed below) and the mean value of the intermediate cluster is intermediate to the means of the two adjoining clusters. The rationale for these three exceptions are:

1) When two clusters have similar means but are separated by a single point that has not been amalgamated into either cluster, the single point may represent a noisy signal that should be eliminated from the clustering process. An exception to this argument is when the clusters are being initially formed; at which point random variation is likely to create this type of pattern. Therefore, criterion (1) is applied only if the combined cluster would have at least ten observations and there are at least two observations in each of the original clusters.

2) Since the clustering algorithm is a stepwise procedure, it need not be optimal in its choice of clusters. Therefore, each time two clusters are merged, it is desirable to check that the cluster was formed from the optimal split into two clusters (as if the clustering algorithm was running in the reverse direction). When the combined cluster can be divided into two adjacent clusters at a different cutpoint for which the within cluster sum of squares is less than the within cluster sum of squares of the original two clusters, the two clusters are realigned by choosing a new boundary that corresponds to the minimum within cluster sum of squares.

3) When two nonadjoining clusters (separated by a single cluster) differ greatly in their mean values ($t_3 > 1.414$ * criterion described below), the intermediate cluster may represent a rise or decline. When the mean of the intermediate cluster is approximately midway (40-60%) between the means of the two neighboring clusters, then the cluster is not included when computing the min t_2 criterion for stopping the clustering algorithm. However, if the stopping criterion is not fulfilled, then the cluster will be combined with its neighbor if t_2 for the cluster is less than min t_2; i.e., these two clusters are identified as the two to be combined.

Step 5:

If the criterion for stopping (discussed below) is fulfilled, then print out the current clusters. Otherwise, return to step 3.

Stopping criterion:

When the usual two-sample t-test is used to test for the equality of levels between clusters, the method of identifying clusters will overidentify the number of clusters because the clusters are chosen to maximize the t-statistic (by combining at each step clusters that minimize the t-like statistic). Therefore, a conservative criterion for cluster identification is desired.

The statistic $t_2 = d/SD$, where SD is the standard deviation of the original sequence, is less than a t-statistic based on the within cluster variance. Therefore, tests based on this statistic will reject the null hypothesis less than tests based on the within cluster variance. An approximate relationship between the two statistics (based on only two clusters in the entire sequence) is:

$$t_2 = \frac{t}{[\,1 + (t^2-1)/(n_1+n_2-1)\,]^{1/2}}$$

where t_2 is the statistic proposed here and t is the pooled two-sample t-statistic. Therefore, t_2 is less than t whenever t is greater than 1. For small sample sizes the disparity may be large; for large sample sizes the disparity will be large only when t is also large (but then t_2 will also be significant.

In our analyses critical values corresponding to 0.05, 0.01 and 0.001 were used. The intermediate value was sufficient to eliminate many small clusters.

The criterion for d_3 was set at 1.414 times that for d_2 in order to adjust for the number of comparisons (searching across three adjacent clusters instead of two).

To stop the clustering algorithm, either each t_2 must be greater than the critical value or the cluster must be contained within a set of three clusters for which t_3 is greater than 1.414 times the critical value. The first and last cluster of the entire sequence do not have to fulfil these criteria in order for stepping to terminate since the sequence may have began or ended during a rise or decline.

EXAMPLES

In Figures 1 and 2 we present the logarithms of the values of luteinizing hormone (LH) in ovariectomized ewes that were sampled twice per week for more than four years. The first ewe (Figure 1) was kept outdoors and therefore subject to an annual photoperiodic stimulus and the second ewe (Figure 2) was kept in a controlled light environment (eight hours light per day) to study the effect of the disruption of the photoperiodic stimulus.

In each figure there are two sets of lines. The uppermost set of lines are the clusters (as identified by the clustering algorithm) plotted at the mean level of hormone for the cluster. The raw data values (on a log scale) are scattered about the cluster lines. In several sections there appear to be more than one cluster (see arrows); the more detailed cluster structure (smaller cluster groups) were obtained using a critical values for t_2 of 1.96 and the longer lines were obtained using a critical value of 2.576.

The lower schematic in each figure presents the results of the regression-like algorithm. Values are graphed on four levels, the lowest is the baseline and highest is the plateau. A cycle would contain both a baseline and a plateau and a return to a baseline. As may be noted, there are many cases of a plateau followed by a rise or decline followed by another plateau (and similarly for baselines). This suggests a change in level of the plateau, but not a full cycle.

Although there are relatively good agreements between the (extended) baselines and plateaus and the clusters, the patterns identified by the regression algorithm appear to be noisier than those identified by the clustering algorithm. However, the clustering algorithm may not identify rises and declines. Also, since the clustering algorithm assumes homogeneous variance on the transformed scale for the entire series, it is less likely to identify cycles whose nadir to peak amplitude is relatively small compared to the overall SD.

The experiment was designed to study whether the annual rhythm is disrupted by removing the photoperiodic stimulus. Note the regularity of the clusters for the control animal (Figure 1) but the change in pattern of the cycles over years under constant light conditions (Figure 2).

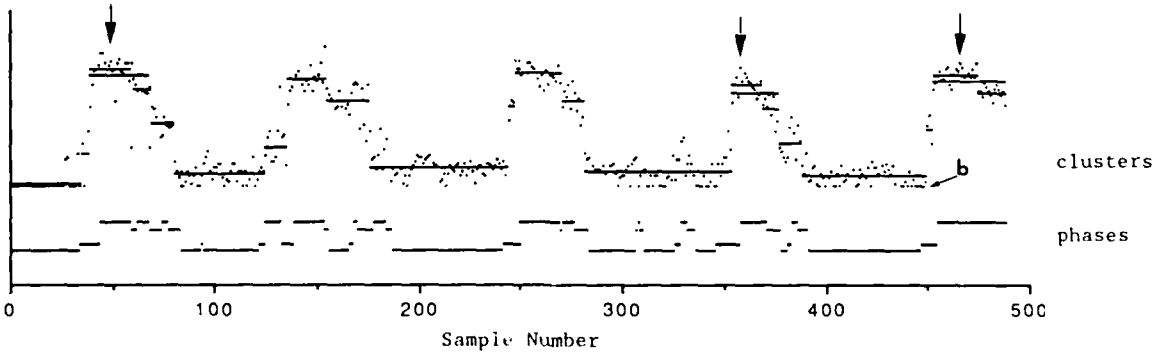


Figure 1. Luteininzing hormone (LH) levels of an ovariectomized ewe (#1006) maintained outdoors. The data points represent levels of LH in blood samples taken twice per week starting on May 24, 1983 and ending on Jan 22, 1988.

a) Data points (on log scale).
b) There is a lower threshhold to the sensitivity of the radioimmunoassay for LH. Therefore, samples at the threshhold appear to form straight lines similar to clusters.
c) Clusters are represented by straight lines at the average level of LH in the cluster. Critical values of 1.96 or of 2.576 were used. Arrows indicate where the solutions differ. The shorter lines are clusters formed by using 1.96. The average level of the two smaller clusters when combined together is equal to the average level of the combined cluster.
d) Phases due to the regression algorithm. The four levels (starting from the top) correspond to the phases: plateau, decline, rise and baseline.

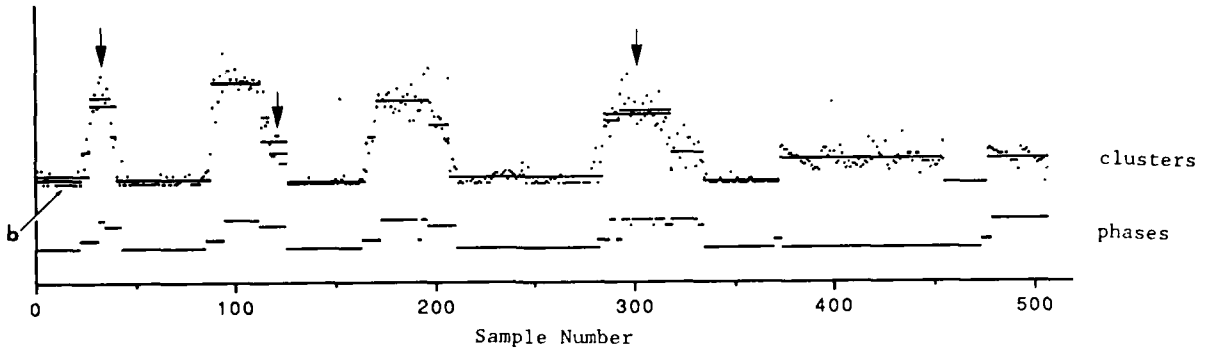


Figure 2. Luteininzing hormone (LH) levels of an ovariectomized ewe (#2021) maintained in a controlled light environment (eight hours of light per day). The data points represent levels of LH in blood samples taken twice per week starting on March 18, 1983 and ending on Jan 22, 1988. See the legend of Figure 1 for explanation of symbols.

DISCUSSION:

We have presented two algorithms to identify phases of a cycle.

The first algorithm identifies all four phases of a cycle; however, it is difficult to formalize statistical tests with respect to individual cycles.

The second algorithm identifies the baseline and plateau. The rises and declines are at times included within the baseline or plateau. An ad hoc test for the presence of a cluster is also presented.

It would appear that a combination of the two algorithms may be preferable to either one individually. For example, the clustering algorithm can be extended so that a straight line is fit to the data between the baseline and plateau (or between the plateau and baseline) so as to best fit (minimize the sum of squares of) the intervening points. To do so, the algorithm might be allowed to include (one or more points adjacent to) the endpoints of the baseline and plateau in the regression calculation. This may improve the estimation of the intermediate phases.

The major advantage of an algorithm is that it will allow comparison among experimental conditions in a consistent manner. Currently, investigators *eyeball* their data to identify different phases. Questionable data points are assigned in a very subjective manner. This algorithm (with additional tuning) should duplicate the investigator's clearcut assignments and then assign the questionable points in a consistent manner.

[1] Morton B. Brown, Department of Biostatistics, 109 South Observatory, The University of Michigan, Ann Arbor, MI 48109-2029

[2] Fred J. Karsch and Benoit Malpaux, Developmental and Reproductive Biology, 300 North Ingalls, 11th floor, The University of Michigan, Ann Arbor, MI 48109

REFERENCES

Bittman, E.L., Dempsey, R.J. and Karsch, F.J. (1983a) Pineal melatonin secretion drives the reproductive response to daylength in the ewe. *Endocrinology* 113, 2276-2283.

Bittman, E.L., Karsch, F.J. and Hopkins, J.W. (1983b) Role of the pineal gland in ovine photoperiodism: regulation of seasonal breeding and the negative feedback effects of estradiol upon luteinizing hormone secretion. *Endocrinology* 113, 329-336.

Box, G.E.P. and Jenkins, G.M. (1976) *Time Series Analysis: Forecasting and Control*. San Francisco, Holden Day.

Farner, D.S. and Follett, B.K. (1979) Repr-ductive periodicitiy in birds. In: *Hormones and Evolution*, pp. 829-872. Ed. E.J.W. Barrington, Academic Press, London.

Koopmans, L.H. (1974) *The Spectral Analysis of Time Series*. Academic Press, New York.

Nett, T.W. and Niswender, G.D. (1982) Influence of exogenous melatonin on seasonality of reproduction in sheep. *Theriogenology* 17, 645-653.

Niswender, G.D., Reichert, L.E. Jr, Midgley, A.R. Jr. and Nalbandov, A.V. (1969) Radioimmunoassay for bovine and ovine luteininzing hormone. *Endocrinology* 84, 1166-1173.

Robinson, J.R. and Follett, B.K. (1982) Photoperiodism in Japanese quail: the termination of seasonal breeding by photorefractoriness. *Proc. R. Soc. Lond. B.* 215, 95-116.

Robinson, J.R. and Karsch, F.J. (1984) Refractoriness to inductive day lengths terminates the breeding season of the Suffolk ewe. *Biol. Reprod.* 31, 656-663.

OPTIMIZATION IN THE DESIGN OF SEQUENTIAL CLINICAL TRIALS

Richard Simon, National Cancer Institute

1. *Introduction*

I use the word "optimization" in the title with some hesitation. "Optimum" clinical trials are those that address an important medical question, obtain a reliable and timely answer and are reported responsibly in the medical literature. I will use the term "optimization" here in a more limited and technical sense to refer to efficiency in the conduct of a clinical trial. I will describe several applications of optimization to the design of simple sequential clinical trials.

2. *Phase II Clinical Trials*

A phase II study of a cancer treatment is an uncontrolled trial for obtaining an initial estimate of the degree of anti-tumor effect of the treatment. The proportion of patients whose tumors shrink by at least 50% is called the response rate and is the statistic of primary interest. The purpose of a phase II trial is to determine whether the drug has sufficient activity against a specific type of tumor to warrant its further development.

The designs described here are based on testing a null hypothesis $H_0: p \leq p_0$ that the true response probability is less than some uninteresting level p_0. If the null hypothesis is true then we require the probability should be less than α of accepting the drug for further study in other clinical trials. We also require that if a specified alternative hypothesis $H_1: p \geq p_1$ that the true response probability is at least some desirable target level p_1 is true then the probability of rejecting the drug for further study should be less than β.

It is rarely practical to utilize a sequential design that requires re-analysis of the data after treatment of each patient. Response assesment may take weeks or months and so the most popular approach to sequential analysis is the "group-sequential" approach in which interim analyses are performed after groups of patients are treated and evaluated. Let n_i denote the number of patients treated in the i'th stage of a phase II trial and let S_i denote the total number of responses observed through the end of the i'th stage. A decision rule or sequential decision boundary may be specified by a set of pairs (l_i, u_i) where we reject the treatment after the i'th stage if $S_i \leq l_i$ and we stop the trial and accept the treatment if $S_i \geq u_i$. Otherwise, the trial proceeds to the i+1'st stage. If we specify the maximum number of stages I and the error limits α and β for the hypotheses based on p_0 and p_1 then one may consider the optimization problem of finding the sample sizes (n_i) and the sequential boundaies (l_i, u_i) for i=1,...,I which satisfy the error probability constraints and minimize

$$\int E[N \mid p]\, w(p)\, df(p).$$

N denotes the number of patients treated in the trial before termination. N is a random variable with maximum value $n_1+\ldots+n_I$. $E[N \mid p]$ denotes the expected value of N when the true response probability is p. The function to be minimized is the expected sample size averaged with regard to a prior distribution f for the unknown response probability p. The average is also weighted by a function w which specifies the relative importance of the sample size for different values of the response probability.

I (Simon 1987) have published designs with 2 stages based on minimizing the expected sample size when $p=p_0$. The designs were limited to two stages because that is often a practical constraint. It may be necessary to stop entering new patients onto a phase II trial at the end of a stage until one determines whether the conditions for continuation have been achieved. Because of the delay in evaluating response, this may require suspension of accrual for weeks or months. Such suspension is an inconvenience to physicians who are deciding how to treat patients and may not be tolerated more than once in the course of a study. I used $w(p_0)=1$ and $w(p)=0$ for all other values of p as a simple way of representing the importance of minimizing the number of patients given an ineffective drug. With this formulation it was not necessary to specify the prior distribution f. Obviously, more general specifications are possible. For many phase II clinical trials, there is no strong desire to terminate early if the treatment appears effective because there are secondary endpoints of interest. If the treatment is inactive, however, the trial should terminate as early as possible. Consequently, I set $u_1=n_1+n_2$ and optimized with regard to the parameters n_1, n_2, l_1 and l_2.

For specified values of p_0, p_1, α and β optimal designs were determined by enumeration using exact binomial probabilities. For each value of total sample size $n=n_1+n_2$ and each value of n_1 in the range $(1, n-1)$ the integer values of l_1 and l_2 were determined which satisfied the two constraints and minimized the expected sample size when $p=p_0$. This was found by searching downward over the range $l_1 \epsilon (0, l_1^*)$; l_1^* is the largest integer for which $B(l_1^*; p_1, n_1) \leq \beta$ where B denotes the cumulative binomial distribution function. For each value of l_1 we determined whether there was a value of l_2 such that the design (n, n_1, l_1, l_2) satisfied both type 1 and type 2 error constraints. If not, then we continued our downward search on l_1. If the design satisfied the constraints, then it was optimal for those values of n and n_1. Keeping n fixed, we searched over the range of n_1 to find the optimal two-stage design for that maximum sample size n. The search over n ranged from a lower value of about

$$\bar{p}(1-\bar{p}) \left[\frac{z_{1-\alpha}+z_{1-\beta}}{p_1-p_0} \right]^2$$

where $\bar{p}=(p_0+p_1)/2$ and the z values are percentiles of the standard normal distribution. We checked below this starting point to ensure that we had determined the smallest sample size n for which there was a nontrivial $(n_1, n_2>0)$ two-stage design which satisfied the error probability constraints. The enumeration procedure searched upwards from this minimum value of n until it was clear that the optimum had been determined. The minimum expected sample size for fixed n is not a unimodal function of n because of discreteness of the underlying binomial distribution. Nevertheless, eventually the local minima increased and a global minimum was identified. Calculations were carried out in APL on a Microvax II computer. Table 1 shows some optimal designs for the case $\alpha=\beta=0.10$. In Table 1 $N_{max} = n_1 + n_2$, $E_0(N)$ is the expected sample size when the null hypothesis is true, and PET_0 is

the "probability of early termination" (after the first stage) when the null hypothesis is true. Designs for large values of p_0 are not usually appropriate for the testing of new drugs but can be useful for pilot studies of combinations of drugs known to be active.

3. Phase II/III Clinical Trials

In cancer therapeutics it is conventional to obtain "promising" data for a new treatment in an uncontrolled phase II study before initiating a randomized comparison to an established regimen. An alternative approach, however, would employ a randomized design from the outset with early termination if preliminary results are not promising for the experimental treatment. Such an approach avoids the difficulties of interpreting results arising from an uncontrolled study.

Thall, Simon, Ellenberg and Shrager(1988a) developed two-stage designs of this type for clinical trials where the endpoint is binary, response or no response. These designs test a null hypothesis $H_0: \Delta = 0$ against a one-sided alternative $H_1: \Delta > 0$ where $\Delta = p_e - p_c$ and p_e , p_c denote the response probabilities for the experimental and control treatments respectively. At the first stage n_1 patients are randomly assigned to receive each of the two treatments. Let r_{i1} denote the observed number of responses with treatment i in the first stage and let $\hat{\Delta}_1 = (r_e - r_c)/n_1$, $\hat{p}_{.1} = (r_{e1} + r_{c1})/2n_1$ and $\hat{q}_{.1} = 1 - \hat{p}_{.1}$. If

$$\frac{\hat{\Delta}_1}{(2\hat{p}_{.1}\hat{q}_{.1}/n_1)^{1/2}} > y_1$$

then continue to the second stage; otherwise terminate the trial and accept H_0.

In the second stage n_2 patients are randomly assigned to receive each of the two treatments. Let $\hat{\Delta}_.$ and $\hat{p}_.$ be defined similarly to the related quantities above but based on all data from the two stages. At the end of the second stage, if

$$\frac{\hat{\Delta}_.}{(2\hat{p}_.\hat{q}_./(n_1+n_2))^{1/2}} > y_2$$

then reject H_0 ; otherwise accept H_0. The constants n_1 , n_2 , y_1 , y_2 are chosen to minimize an average expected sample size subject to error probability constraints. The probability of rejecting H_0 should be no greater than a specified α whenever the null hypothesis is true, regardless of what the common value of the response probability is. The probability of rejecting the null hypothesis should be at least $1 - \beta$ for a specified alternative $\theta = (p_e, p_c)$. We minimized the expected sample size averaged equally over H_0 and over θ. Minimization of $E_\theta(N)$ alone produces designs with low probability of stopping at the end of the first stage and hence relatively poor performance when the null hypothesis is true. Minimization of $E_0(N)$ alone produces designs with high probabily of early termination under H_0 but large maximum sample sizes, and consequently poor performance under the alternative. Although we could have minimized relative to a prior and weight function on the space of (p_e, p_c), the simple approach that we used resulted in non-extreme designs with generally good performance under both the null and alternative hypotheses. Performance under the alternative hypothesis could be improved by permitting rejection of the null hypothesis after the first stage. Thall et. al. (1988a) show how this can be accomplished by superimposing an early rejection rule of the O'Brien-Fleming (1979) type.

The optimization problem was solved on a DEC-10 computer using MLAB, an interactive mathematical modeling program with built-in integration and curve fitting capabilities (Knott 1979). For each selected pair (n_1, n_2), we determined values of y_1 and y_2 by solving nonlinear equations representing the error constraints. Zeros of the two equations were determined by solving a non-linear regression problem. We used normal approximations to the binomial distribution and performed numerical integration using a variable step Adams-Moulton predictor-corrector method. We determined the optimum design by a systematic search of an integer grid of (n_1, n_2) values.

Table 2 shows some of the resulting optimal designs for α=0.05 and β=0.20. The column labled N_{fixed} represents the size of a single stage design with the same error probabilities for the null and specified alternative hypotheses. The large reduction in expected sample size under the null hypothesis is obtained with very little increase in maximum sample size compared to the fixed sample design. The column labled PET_0 represents the probability of early termination after the first stage when the null hypothesis is true.

4. Two-Stage Selection and Testing Designs

In clinical research there are often several experimental treatments of interest but too few patients available to thoroughly evaluate each relative to a control therapy. A common approach in such circumstances is to first select the experimental treatment which appears most promising based on uncontrolled pilot studies and then compare the selected treatment to the control in a large randomized clinical trial. When such pilot studies are performed at different institutions, treatment effects typically are confounded with other factors and the selection of a most promising regimen is problematic. Thall, Simon and Ellenberg (1988b) proposed a new approach to the problem of identifying the best of several experimental treatments and determining whether it is superior to a control. We developed a two-stage design for use with binary endpoints.

During the first stage n_1 patients are randomly assigned to each of the K experimental treatment groups and n_1 patients are randomly assigned to the control group. At the end of the first stage the largest observed response rate for the experimental treatments is compared to the observed response rate for the control group. If the standardized normal Z value for that comparison does not exceed a critical value y_1 , then the clinical trial is terminated and no experimental treatment is claimed to be better than the control. Otherwise, a second stage is conducted in which an additional n_2 patients are assigned to the control treatment and to the experimental treatment with the greatest

response rate in the first stage. Thus, at most one experimental treatment is carried over to the second stage. The treatment carried into the second stage is called the "selected" treatment. At the end of the second stage the selected experimental treatment is compared to the control based on all data for those two treatments obtained in either stage. If the standardized normal Z value for that comparison exceeds a critical value y_2, then the global null hypothesis of equivalence of all the treatments is rejected and the selected experimental treatment is "chosen" as more effective than the control. Otherwise, the global null hypothesis of the equivalence of all K+1 treatments is not rejected.

As for the phase II/III design described above, our intent was to determine the parameters n_1, n_2, y_1, y_2 to minimize an average expected sample size subject to constraints on the type 1 and type 2 errors. The type 1 error constraint is straightforward, the probability of rejecting the global null hypothesis when it is true should not exceed a specified α, whatever the value of the common response probability. The nature of the type 2 error, or power, constraint was much more complicated, however, because of the great variety of alternative hypotheses possible with K+1 treatments. We specified a generalized power constraint in the following way. An experimental treatment whose response probability exceeded that of the control by at least a pre-specified quantity δ_2 is called "effective". An experimental treatment whose response probability exceeded that of the control by more than δ_1 but by less than δ_2 is called "marginally effective". We require that if there is at least one effective treatment and no marginally effective treatments then the probability of choosing an effective treatment as better than the control must be at least $1-\beta$. If there are marginally effective treatments that are almost as good as the effective treatments, then there will be a substantial probability that one of them will be selected and chosen instead of an effective treatment, but the difference is of little consequence. If there are marginally effective treatments that are much worse than the effective treatments, they will have little influence on the probability of choosing an effective treatment. The least favorable configuration for this constraint is that with one experimental treatment having response probability exactly δ_2 greater than the control and the remaining K-1 experimental treatments having response probabilities exactly δ_1 greater than the control.

We determined the values of the design parameters to minimize the average expected sample size, weighted equally between the null hypothesis and the least favorable alternative configuration, subject to the type 1 error constraint and the generalized power constraint. The optimization algorithm was based on a grid search over n_1 and y_1. An integer grid was used for the former and a grid width of 0.025 for the latter. For specified values of n_1 and y_1, the nonlinear equations for type 1 error and generalized power were solved for the parameters y_2 and $\pi=n_1/(n_1+n_2)$. Those equations were solved to an accuracy of $\pm 10^{-4}$ using the least squares algorithm of Shrager(1970). Regarded as a function of y_1 for fixed n_1, the average expected sample size had two distinct local minima in all cases. A finer grid search in the neighborhoods of these local minima was carried out to obtain the minimum given n_1. As a function of n_1, this minimum is unimodal, thus yeilding the global optimum.

Table 3 shows some of the optimum designs determined for $\alpha = 0.05$, $\beta = 0.25$. The probabilities of early termination under the null hypothesis are generally in excess of 0.50 and the maximum sample sizes are less than single stage trials with similar design objectives, such as those of Dunnett(1984).

5. Conclusion

I have presented some of the research that I and my colleagues have conducted in the past few years in the area of optimized sequential designs for clinical trials. I have not attempted to present a review of related work by others although this topic is one of increasing interest on the part of biostatisticians. Although there is a great literature on sequential designs for clinical trials, it is only recently that these methods have seen broad application. Clinical trials are complex endeavors and the simplest designs are often the most practical. For this reason we have focused on two-stage designs. Even with such simple designs it is possible to achieve substantial reductions in required sample size compared to single stage designs. Such reductions translate into reduced exposure of patients to ineffective treatments and increased efficiency in the process of discovering effective ones.

6. References

Knott GD (1979). MLAB: A mathematical modeling tool. *Computer Programs in Biomedicine* 10:271-280.

Dunnett CW (1984). Selection of the best treatment in comparison to a control with application to a medical trial. In *Design of Experiments: Ranking and Selection*. Santner TJ and Tamhane AC eds, 47-66, New York: Marcel Dekker.

O'Brien PC, Fleming TR (1979). A multiple testing procedure for clinical trials. *Biometrics* 35:549-556.

Shrager RI (1970). Nonlinear regression with linear constraints: an extension of the magnified diagonal method. *Journal of the Association for Computing Machinery* 17:446-452.

Simon R (1987). How large should a phase II trial of a new drug be? *Cancer Treatment Reports* 71:1079-1085.

Thall P, Simon R, Ellenberg SS, Shrager R (1988a). Optimal two-stage designs for clinical trials with binary response. *Statistics in Medicine* 7:571-579.

Thall P, Simon R, Ellenberg SS (1988b). Two-stage selection and testing designs for comparative clinical trials. *Biometrika* (in press).

TABLE 1. Phase II Designs for $\alpha = \beta = 0.10$

p_0	p_1	Reject Drug If Response Rate $\leq l_1/n_1$	$\leq l_2/N_{max}$	$E_0(N)$	PET_0
0.10	0.30	1/12	5/35	19.8	0.65
0.20	0.40	3/17	10/37	26.0	0.55
0.30	0.50	7/22	17/46	29.9	0.67
0.40	0.60	7/18	22/46	30.2	0.56
0.50	0.70	11/21	26/45	29.0	0.67
0.60	0.80	6/11	26/38	25.4	0.47
0.70	0.90	6/9	22/28	17.8	0.54

TABLE 2. Two-stage phase II/III designs for $\alpha = 0.05$ $\beta = 0.20$

p_c	p_e	n_1	n_2	y_1	y_2	$E_0(N)$	PET_0	N_{max}	N_{fixed}
0.20	0.40	28	40	0.32	1.59	86.0	0.63	136	128
0.30	0.50	33	45	0.36	1.58	98.5	0.64	156	148
0.40	0.60	33	49	0.34	1.58	102.0	0.63	164	154
0.50	0.70	33	45	0.36	1.58	98.5	0.64	156	148
0.60	0.80	28	40	0.32	1.59	86.0	0.63	136	128
0.70	0.90	21	31	0.35	1.58	64.4	0.64	104	98

TABLE 3. Two-stage selection and testing designs for $\alpha = 0.05$ $\beta = 0.25$

K	p_c	n_1	n_2	y_1	y_2	$E_0(N)$	PET_0	N_{max}
2	0.20	36	44	0.73	1.82	139.7	0.64	196
2	0.40	40	58	0.59	1.81	169.3	0.57	236
2	0.60	31	50	0.54	1.80	134.6	0.58	193
3	0.20	38	59	0.71	1.92	205.3	0.55	270
3	0.40	47	63	0.55	1.94	254.9	0.47	314
3	0.60	37	55	0.51	1.94	204.1	0.49	258
4	0.20	44	62	0.87	1.98	271.7	0.58	344
4	0.40	49	77	0.55	2.00	336.8	0.40	399
4	0.60	42	61	0.72	2.00	268.6	0.52	332

BAYES ESTIMATION OF CEREBRAL METABOLIC RATE OF GLUCOSE IN STROKE PATIENTS

P. David Wilson, University of South Florida College of Public Health
Sung-Cheng Huang and Randall A. Hawkins, UCLA School of Medicine

Local cerebral metabolic rate of glucose (LCMRG) is defined as a nonlinear function of the rate constants in a three-compartment model. Data for estimating LCMRG in the human brain is obtained by PET scanner following injection of F-18 labeled fluorodeoxyglucose. Optimal analysis would be based on scans repeated up to three hours, but this is not practical. Nuclear medicine scientists have therefore developed three single scan (SS) methods requiring only a single ten minute scan taken at about one hour post-injection. These SS methods use prior information in the form of mean rate constants from the normal (healthy) population, but are not Bayes methods. We have developed a Bayes method which can be used with a single scan. For brains of stroke patients (which contain mostly normal tissue and some ischemic tissue), the Bayes method uses a highest posterior density criterion to choose between prior densities from normal and ischemic tissue populations. Computer simulation studies show that the Bayes SS method is superior to the non-Bayes SS methods.

KEY WORDS: Bayes Estimation, Compartmental models, Glucose metabolic rate.

1. INTRODUCTION

The current method for measurement of local cerebral metabolic rate of glucose (LCMRG) utilizes positron emission tomography (PET) images of the concentration of F-18 (a positron-emitting isotope of fluorine) in a local region of brain tissue, obtained after intravenous injection of F-18 labeled fluorodeoxyglucose (FDG), and while LCMRG is in steady-state.

Analysis of PET data is based on the three-compartment model for FDG kinetics shown in Figure 1. FDG is injected into the plasma compartment, from which it communicates with the brain tissue compartments. Once in the tissue, FDG can undergo phosphorylation, the first step in glucose metabolism. From the plasma compartment, FDG is also lost to urine and other tissues (not shown in Figure 1).

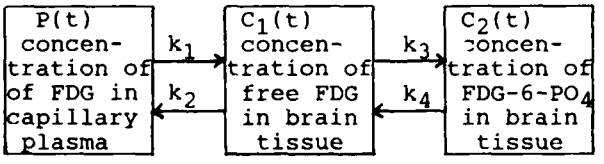


Figure 1. Compartmental Model for FDG Kinetics. $k_1,\ldots,k_4$ are the FDG rate constants. $C(t) \equiv C_1(t) + C_2(t)$.

Unlike glucose, FDG does not proceed further down the metabolic path. This allows sufficient accumulation of FDG in brain tissue to provide relatively precise positron emission count statistics. It also prevents recirculation of any metabolic products containing F-18, which would contaminate the plasma compartment. LCMRG is defined as the net rate of phosphorylation of glucose. However the rate constants for glucose are not the same as those for FDG, and it has been shown that

$$LGMRG = (P_G/LC)k_1k_3/(k_2 + k_3) \qquad (1)$$

where P_G is the capillary plasma concentration of glucose (required to be in steady state) and LC is a "lumped constant". For our purposes LC may be taken to be a known constant which accounts for the use of FDG rate constants instead of glucose rate constants.

2. MATHEMATICAL MODEL

With respect to the compartmental model, measurements of P(t) are obtained by repeated sampling of a peripheral vessel. The PET scanner provides a noisy version of C(t), the total F-18 concentration in the brain, defined as $C(t) = C_1(t) + C_2(t)$ in the compartmental model. As an approximation, the contribution to the PET data from the FDG in the brain capillaries is usually ignored. (However a more general formulation including this contribution can be found in Hawkins, Phelps, and Huang, 1986.)

From a linear systems viewpoint, P(t) can be viewed as the input function to a linear system with output function C(t) and impulse response $h(t;\underline{k})$, described below, where $\underline{k}$ is the set of rate constants. The differential equations implied by the compartmental model are

$$dC_1(t)/dt = k_1P(t) + k_4C_2(t) - (k_2+k_3)C_1(t)$$

$$dC_2(t)/dt = k_3C_1(t) - k_4C_2(t) \qquad (2)$$

where P(t) is treated as a known (measured) function. To conveniently express the solution of equations (2) we define "macroparameters", $\underline{a} = (a_1, a_2, a_3, a_4)$ as follows:

$$a_4, a_2 = [k_2+k_3+k_4 \pm \{(k_2+k_3+k_4)^2 - 4k_2k_4\}^{\frac{1}{2}}]/2$$

$$a_1 = k_1(k_3+k_4-a_2)/(a_4-a_2)$$

$$a_3 = k_1(a_4-k_3-k_4)/(a_4-a_2). \qquad (3)$$

Then

$$C_1(t) = [k_1/(a_4-a_2)][(k_4-a_2)\exp(-a_2t) + (a_4-k_4)\exp(-a_4t)]\circledast P(t)$$

$$C_2(t) = [k_1k_3/(a_4-a_2)][\exp(-a_2t)-\exp(a_4t)]\circledast P(t)$$

$$C(t) = C_1(t) + C_2(t) = [a_1\exp(-a_2t) + a_3\exp(-a_4t)]\circledast P(t) \quad (4)$$

where $\circledast$ denotes convolution from 0 to t. The impulse response expressed earlier as $h(t;\underline{k})$ can be defined in terms of $\underline{a}$ as

$$h(t;\underline{a}) = a_1\exp(-a_2t) + a_3\exp(-a_4t) \quad (5)$$

so that the output function is

$$C(t) = h(t;\underline{a})\circledast P(t). \quad (6)$$

Equation (6) is the mathematical model for the expected value of the PET observations.

3. EXISTING ESTIMATION PROCEDURES

3.1 Direct Method

For research purposes, usual nonlinear regression methods are used to estimate the rate constants, $\underline{k}$, or the macroparameters, $\underline{a}$. If the macroparameters are estimated, inversion of equations (3) yields estimates of the rate constants. Then equation (1) is used to estimate LCMRG.

The PET scan data collection scheme usually consists of ten 2-minute scans followed by ten 5-minute scans followed by ten or eleven 10-minute scans, for a total scan time of approximately three hours. The rapid scans at first are required to record the rapidly changing brain concentration of F-18 immediately after injection. Later, as the brain concentration changes more slowly, longer duration scans are used to compensate for loss of precision due to decay of F-18, which has a physical half-life of approximately two hours. The long total scan time is required because a_2 is usually on the order of 10^{-3}.

Measurements of P(t) are taken from a peripheral vessel beginning immediately after injection. P(t) rises extremely rapidly to reach a sharp peak, usually within the first minute, and then falls rapidly at first before beginning a more gradual decline after about ten minutes. Samples are usually taken at 5 to 10 second intervals for the first 3 minutes and then at progressively lengthening intervals for the remainder of the 3 hour study. The samples are counted externally in a well counter to determine F-18 activity and are calibrated relative to the PET observations.

The P(t) data are generally quite noise-free. However if smoothing is required, a nonparametric smoothing algorithm such as found in Wilson (1988) can be used to smooth from the peak to the end of the study. Samples of P(t) are generally sufficiently closely-spaced so as to allow a rather accurate numerical representation of P(t) by simple linear interpolation between sampling times. Convolution of P(t) with exp(-at) can then be performed by analytically convolving the straight line segments with the exponential function. Convolution is required up to each brain sampling time.

The direct method is recognized to provide accurate estimates of LCMRG, conditional on the given value of the lumped constant, LC, and is the method of choice for research. For routine clinical studies, however, the direct method is impractical because of the three-hour scanning requirement. Demand for scanner time dictates shorter duration studies. Furthermore the difficulties of keeping the patient's head immobilized in the scanner for three hours cannot generally be overcome in a routine clinical setting. For these reasons nuclear medicine scientists have developed several methods which require only a single PET scan of duration no greater than 10 minutes. These are discussed next.

3.2 Non-Bayes Single-Scan Methods

The three single scan (SS) methods described in this section all use prior information but are not Bayes procedures. They are distinguished from the Bayes SS method which we developed, and which is described in the next section. All four SS methods require only a single PET scan, of duration usually 10 minutes and centered at time t = T, which is usually 40 to 60 minutes post-injection. All of the SS methods make use of estimates of the mean values of the rate constants or macroparameters in the normal population. These estimates are available from studies employing the direct method. (See Huang and Phelps, et al, 1980.)

Let $\bar{\underline{k}} = (\bar{k}_1, \bar{k}_2, \bar{k}_3, \bar{k}_4)$ be the estimates of the normal population mean rate constants. Let LCMRG($\bar{\underline{k}}$) be LCMRG of equation (1), evaluated at $\bar{\underline{k}}$. Let $C(T;\bar{\underline{k}})$, $C_1(T;\bar{\underline{k}})$, and $C_2(T;\bar{\underline{k}})$ be C(t), $C_1(t)$, and $C_2(t)$ of equations (4) evaluated at t = T and $\underline{k} = \bar{\underline{k}}$, and with use of P(t) from the subject under measurement.

Let y(T) be the PET scan measurement of the subject at t = T: y(T) = C(T) + noise. The first non-Bayes SS method for estimating LCMRG is due to Sokoloff, Phelps, and Huang, and the estimator is denoted herein as LCMRG(SPH):

$$\text{LCMRG(SPH)} = \text{LCMRG}(\bar{\underline{k}})[y(T)-C_1(T;\bar{\underline{k}})]/C_2(T;\bar{\underline{k}}). \quad (7)$$

Note that if $y(T)$ were replaced with $C(T;\underline{\bar{k}})$ in equation 7, LCMRG(SPH) would simply be LCMRG($\underline{\bar{k}}$).

All of the foregoing development in sections 1, 2, and 3 can be found in greater detail for the biomedical reader in Phelps and Huang et al (1979), and Huang and Phelps et al (1980).

Let $\underline{\bar{a}} = (\bar{a}_1, \bar{a}_2, \bar{a}_3, \bar{a}_4)$ denote the estimates of the normal population mean macroparameters, (obtained from the same source as $\underline{\bar{k}}$). The second non-Bayes SS method is due to Brooks (1982), and the estimator is denoted herein as LCMRG(B):

$$LCMRG(B) = LCMRG(\underline{\bar{k}})[y(T)-\bar{a}_3\exp(-\bar{a}_4T) \circledast P(T)]/[\bar{a}_1\exp(-\bar{a}_2T)\circledast P(T)] \quad (8)$$

where $P(t)$ is from the patient under measurement. Note, again, that if $y(T)$ were replaced with $C(T;\underline{\bar{a}})$, LCMRG(B) would simply be LCMRG($\underline{\bar{k}}$).

The third SS method for estimating LCMRG is due to Hutchins and Holden et al (1984), and the estimator is

$$LCMRG(H) = LCMRG(\underline{\bar{k}})y(T)/[C_1(T;\underline{\bar{k}}) + C_2(T;\underline{\bar{k}})]. \quad (9)$$

As before, if $y(T)$ were replaced with $C(T;\underline{\bar{k}})$, then LCMRG(H) = LCMRG($\underline{\bar{k}}$). Hutchins and Holden et al point out that this estimator is independent of $\bar{k}_1$ since LCMRG($\underline{\bar{k}}$), $C_1(T;\underline{\bar{k}})$, and $C_2(T;\underline{\bar{k}})$ are all linear in $\bar{k}_1$, and the term cancels in their estimator. They argue that this should be of value in studying ischemic tissue of stroke patients since k_1 will be diminished in such tissue.

For our purposes it is important to point out that Hawkins, Phelps, Huang, and Kuhl (1981) studied the behavior of LCMRG(SPH) in normal tissue and the ischemic tissue of stroke patients. They found that while LCMRG(SPH) behaved reasonably well in normal tissue, it had a negative bias of about 50% in ischemic tissue. This finding was confirmed in simulation studies by Wilson, Huang, and Links (1984), who also studied LCMRG(B) by simulation, and found it to have a negative bias of about 35 to 40% in ischemic tissue. Those authors also showed, by simulation, that if $\underline{\bar{k}}$ or $\underline{\bar{a}}$ from an ischemic tissue population is used by LCMRG(SPH) and LCMRG(B) when studying ischemic tissue, these SS estimators behave quite well.

The non-Bayes SS methods must use prior information, $\underline{\bar{k}}$ or $\underline{\bar{a}}$, from a specified population. Although a portion of the brain of a stroke patient is ischemic, perfusion in most of the brain is normal. In studying such a brain with a non-Bayes SS method there are two choices: (1) use prior information from the normal population, or (2) perform preliminary perfusion scans to determine the perfusion status of the various regions and use prior information from the normal or ischemic tissue population in studying the LCMRG of a given region, according to the perfusion status found for that region. We defined the non-Bayes SS methods as using prior information from the normal population because that is the way they are usually used clinically. The preliminary perfusion scan is usually avoided because of the additional scanner time, the additional radiation dose to the patient, the additional effort required in evaluating the scan, and the delay this causes in estimating the LCMRG.

We have developed a Bayes procedure which can be used as a SS procedure, and which can choose between estimates of LCMRG based on prior information from two sources (normal and ischemic tissue populations in our case) using a highest posterior density criterion unavailable to the non-Bayes methods. The Bayes procedure is described next.

4. BAYES ESTIMATION

Although, formally, the Bayes estimates of $\underline{a}$ should be converted to rate constant estimates (using the inverse of equations (3)), and these estimates then used in equation (1), we found empirically that this procedure has some negative bias which can be partly eliminated by the estimator

$$LCMRG(Bayes) = (P_G/LC)\hat{a}_1 \quad (10)$$

where $\hat{a}_1$ is the Bayes estimator of macroparameter a_1. The justification for this choice is as follows: Brooks (1984) pointed out that $a_1 = \beta R$, where $R = k_1k_3/(k_2 + k_3)$, and $\beta \to 1$ as $k_4 \to 0$. In the data base used to provide estimates $\underline{\bar{k}}$ and $\underline{\bar{a}}$ for prior information (Huang and Phelps et al), we found that $\beta \approx 1.05$ with very little variation among individuals. Thus using $\hat{a}_1$ as the estimator of R compensates for some of the negative bias which would otherwise occur.

Let the set of mid-scan times be t_i, $i=1,\ldots,n$. Although we emphasize the use of Bayes Estimation as a SS procedure, we describe the general procedure. Re-express $C(t_i)$ of equation (4) as $C(t_i;\underline{a})$ and shorten it to $C_i(\underline{a})$. Let y_i be the PET scan observation at time t_i so that $E(y_i) = C_i(\underline{a})$. Let $\underline{y} = (y_1, y_2, \ldots, y_n)'$ (where prime denotes transposition), and define $\underline{a} = (a_1, a_2, a_3, a_4)'$ to be the column vector of the macroparameters. Let $\Theta \equiv \Theta(\underline{a},\underline{y})$ denote the true sum of squared errors: $\Theta = \Sigma_{i=1}^{n}[y_i-C_i(\underline{a})]^2$. Let the variance of y_i be v, assumed here to be constant over i. Let $\tau = 1/v$ be the "precision". (It is more convenient to use a prior density for τ than for v). The density of the data is assumed to be Gaussian:

$$f_Y(\underline{y}|\underline{a},\tau) \propto \tau^{n/2}\exp(-\tau\Theta/2). \quad (11)$$

The prior density of $\underline{a}$ is assumed to be Gaussian with mean vector $\underline{a}_o$, covariance matrix Ω, and independent of τ:

$$f_a(\underline{a}|\underline{a}_o,\Omega) \propto |\Omega|^{-\frac{1}{2}} \exp\{-(\underline{a}-\underline{a}_o)'\Omega^{-1}(\underline{a}-\underline{a}_o)/2\}. \quad (12)$$

The prior density of τ is assumed to be Gamma with parameters δ and μ so that $E(\tau) = \delta/\mu$ and $var(\mu) = \delta/\mu^2$:

$$f_T(\tau|\delta,\mu) \propto \tau^{\delta-1}\exp(-\mu\tau), \text{ for } \tau,\mu,\delta > 0. \quad (13)$$

(See pp 3-5 of Broemeling, 1985, and pp 77-82 of Vinod and Ullah, 1981. In the treatment of those authors, Ω in equation (12) is replaced with Ω/τ. This is of no use here because our empirically estimated prior moments are $\underline{a}_o$ and Ω, whereas τ is unknown.)

The joint posterior density of $\underline{a}$ and τ is proportional to the product of the right hand sides of equations (11), (12), and (13). After integrating τ out of this joint posterior density, one obtains the marginal posterior density of $\underline{a}$:

$$p(\underline{a}|\underline{y},\underline{a}_o,\Omega,\delta,\tau) \propto f_a(\underline{a}|\underline{a}_o,\Omega)[2\mu+\theta]^{-(\delta+n/2)}. \quad (14)$$

We defined our Bayes estimator of $\underline{a}$ in terms of maximum posterior density (*MPD*) estimation for computational simplicity rather than using minimum-Bayes-risk estimation (Bard, 1974, pp 61-75). To maximize the posterior density in equation (14), one must solve, for k = 1,2,3,4,

$$\Sigma_{i=1}^{n}[\dot{C}_{ki}(\underline{a})\{y_i-C_i(\underline{a})\}]-[(\theta+2\mu)/(n+2\delta)]$$
$$[\Sigma_{j=1}^{4}(a_j-a_{oj})u_{kj}+(a_k-a_{ok})u_{kk}]/2 = 0 \quad (15)$$

where $\dot{C}_{ki}(\underline{a}) = \partial C_i(\underline{a})/\partial a_k$, u_{ij} is defined by $\Omega^{-1} = U = (u_{ij})$, and a_{ok}, k = 1,...,4, are the elements of $\underline{a}_o$.

The solution of equations (15) is obtained twice: once using the prior moments $\underline{a}_o$ and Ω from the normal tissue population and once using these prior moments from the ischemic tissue population. The solution producing the highest posterior density in equation (14) is chosen as the Bayes estimate of $\underline{a}$.

5. COMPUTER SIMULATION STUDIES COMPARING BAYES AND NON-BAYES SS METHODS IN ISCHEMIC TISSUE

Small data bases of rate constants estimated by the direct method have been published by Huang and Phelps et al (1980) for normal tissue, and by Hawkins, Phelps, Huang, and Kuhl (1981) for ischemic tissue. We consider only gray matter tissue here. While these data bases are too small to provide prior moments for empirical Bayes analyses of actual human data, they can nevertheless form the bases for "simulation populations" for comparing the behavior of the four SS estimators described above.

To allow for some modeling error in the prior distribution of $\underline{a}$, and at the same time rule out any negative elements of $\underline{a}$ in the simulation population, we assumed the elements of $\underline{a}$ to be distributed joint lognormal in the simulation population. Let $\underline{L}$ and S be the mean and covariance matrix, respectively, of the logs of the elements of $\underline{a}$ in the simulation population. These moments were taken to be the values computed from the logs of the elements of the $\underline{a}$-estimates in the data base. To obtain the $\underline{a}$ for a simulated subject, we generated a pseudo-random realization of a four-variate Gaussian random vector with moments $\underline{L}$ and S, and then exponentiated the elements. The macroparameters for all simulated subjects were generated from the ischemic simulation population.

After generating the macroparameters, $\underline{a}$, for a simulated subject, the impulse response, $h(t;\underline{a})$ in equation (5), was generated. A plasma curve, P(t), for the individual was then generated as a combination of 5 exponentials with coefficients randomly selected from ranges seen in practice, and constrained so that $C(t) = h(t;\underline{a})\circledast P(t)$ rises over the first hour to match clinical experience. The single scan time chosen was T=1 hour. The PET data, y(T), was then created as $y(T) = C(T) + \varepsilon$, where ε was a pseudo-random realization of a zero-mean Gaussian variate with standard deviation 0.05 C(T). The multiple 0.05 was chosen because in the ischemic data base, in which every $\underline{a}$-estimate was accompanied by its associated mean-squared-error (MSE) of fit, the average root MSE was approximately 5% of the fitted value of C(T). The data, {y(T), P(t)} were then analyzed by each of the four SS methods.

The factor P_G/LC was not used in estimating LCMRG because all results were recorded as percent error, and P_G/LC is a common multiplier in both the true LCMRG and all four estimators.

Prior moments $(\underline{a}_o,\Omega)$ were available from both the normal and the ischemic simulation populations for use in MPD Bayes estimation. The prior mean from the normal population was used in the three non-Bayes SS methods. The values of μ and δ used in the Bayes estimation were obtained as follows: Letting m and d be, respectively, the mean and variance of the reciprocal MSE values in the data base, we solved the equations $m = \delta/\mu$ and $d = \delta/\mu^2$ for μ and δ.

The simulation studies were designed to show certain characteristics of the distribution of percent errors of the four estimators as a function of the true

value of $R = k_1k_3/(k_2+k_3)$. We simulated 100 sets of data in each of 8 intervals in R from 0.01 to 0.03 with interval width 0.0025. In generating the data for a particular interval in R, macroparameters with associated value of R not in the interval were discarded, and generation continued until 100 sets of macroparameters with R in the specified interval were obtained.

Results of the simulation studies are shown in Figures 2, 3, and 4. Figure 2 shows the root-mean-square of the distribution of percent errors in the 100 analyses by each of the four estimators in each of the 8 intervals in R. Values of R in the last two intervals on the right are in the low normal range (even though all macroparameters were generated from the ischemic simulation population). Because the Bayes and Hutchins-Holden procedures are distinctly superior to the other two estimators, Figures 3 and 4 display behavior of only these two superior estimators.

Figure 3 shows the mean of the distribution of percent errors of the same 100 analyses by the Bayes and Hutchins-Holden methods in each interval of R. This figure shows that most of the inferior behavior of the Hutchins-Holden estimator is due to a negative bias of about 12% on average.

Figure 4 shows the range in the distribution of percent errors. In the range $0.0125 \leq R \leq 0.0275$, the largest absolute percent errors were smallest for the Bayes procedure.

These results indicate, as expected, that the Bayes SS procedure should outperform the three non-Bayes SS procedures in analysis of actual human data, once a sufficiently large data base becomes available so that it can serve as the basis for empirical prior moments. It is a tribute to the Hutchins-Holden procedure that it performs as well as it does. Because it is computationally much less burdensome than the Bayes procedure, LCMRG(H) presents a challenge to statisticians to develop a simpler procedure which can outperform it.

A report of this work for biomedical readers can be found in Wilson, Huang, and Hawkins (1988).

ACKNOWLEDGEMENTS

This work was supported by grant #NS22474 from the National Institute of Neurological and Communicative Disorders and Stroke, and by the University of Maryland Computer Science Center. The work was done while the first author was at the University of Maryland School of Medicine.

REFERENCES

Bard Y (1974). Nonlinear Parameter Estimation. Academic Press, New York.

Broemiling LD (1985). Bayesian Analysis for Linear Models. Marcel Dekker, New York.

Brooks RA (1982). Alternative formula for glucose utilization using labeled deoxyglucose. J. Nuc. Med. 23:538-539.

Hawkins RA, Phelps ME, Huang SC, Kuhl DE (1981). Effect of ischemia on quantification of local cerebral glucose metabolic rate in man. J.Cereb. Blood Flow Metab., vol 1, #1,pp 37-51.

Hawkins RA, Phelps ME, and Huang SC (1986). Effects of Temporal Sampling, Glucose Metabolic Rates, and Disruptions of the Blood-Brain Barrier on the FDG Model With and Without a Vascular Compartment: Studies in Human Brain Tumors With PET. J. Cereb. Flow Metab., vol 6, #2, pp 170-183.

Huang SC, Phelps ME, Hoffman EJ, Kuhl DE (1981) Error sensitivity of fluorodeoxygluxose method for measurement of cerebral metabolic rate of glucose. J. Cereb. Blood Flow Metab., vol 1, #4 pp 391-401.

Huang SC, Phelps ME, Hoffman EJ, Sideris K, Seliln CJ, Kuhl DE (1980). Noninvasive determination of local cerebral metabolic rate of glucose in man. Am. J. Physiol. 238:E69-E82.

Hutchins GD, Holden JE, Koeppe RA, Halama JR, Gately SJ, Nickles RJ (1984). Alternative Approach to Single Scan Estimation of Cerebral Glucose Metabolic Rate Using Glucose Analogs, With Particular Attention to Ischemia. J. Cereb. Blood Flow Metab., vol 4, #1, pp 35-40.

Phelps ME, Huang SC, Hoffman EJ, Selin C., Sokoloff L, Kuhl JDE (1979). Tomographic measurement of local cerebral glucose metabolic rate in humans with (F-18) 2-fluoro-2-deoxy-D-glucose: Validation of the method. Ann. Neurol. 6:371-388.

Sokoloff L, Reivich M, Kennedy C, Des Rosires MH, Patlak CS, Pettigrew KD, Saekurada M, Shinohara M (1977). The (C-14) deoxyglucose method for measurement of local glucose utilization: theory, procedure, and normal values in the conscious and anesthetized albino rat. J. Neurochem. 28:897-916.

Vinod HD and Ullah A (1981). Recent Advances in Regression Methods. Marcel Dekker, New York.

Wilson PD, Hawkins RA, Huang SC (1986). Bayes Regression Computation of Local Cerebral Metabolic Rate of Glucose in Stroke. Abstract. J. Nucl. Med. Vol 27, p. 1004.

Wilson PD, Huang SC, Links JM (1984). Improved Estimation of Local Cerebral Glucose Metabolic Rate Using Bayes Regression Analysis of PET Scan Data. Proceedings, Eighth Annual Symposium on Computer Applications in Medical Care. IEEE Computer Society Press,pp 128-131.

Wilson PD, Huang SC, Hawkins RA (1988). Single Scan Bayes Estimation of Cerebral Glucose Metabolic Rate: Comparison With Non-Bayes Single Scan Methods Using FDG PET Scans in Stroke. J. Cereb. Blood Flow Metab., vol 8, #3 pp 418-425.

Wilson PD (1988). Autoregressive Growth Curves and Kalman Filtering. Statistics in Medicine. Vol 7, #1/2, pp 73-86. (See section 5.1).

FIGURE 1. ROOT MEAN SQUARE PERCENT ERRORS

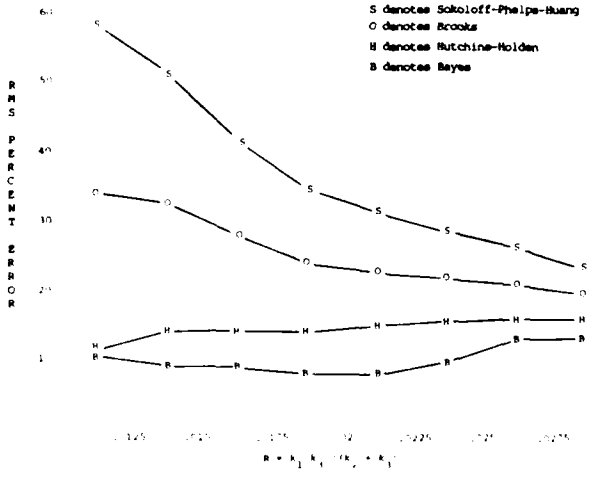


FIGURE 2. MEAN PERCENT ERRORS

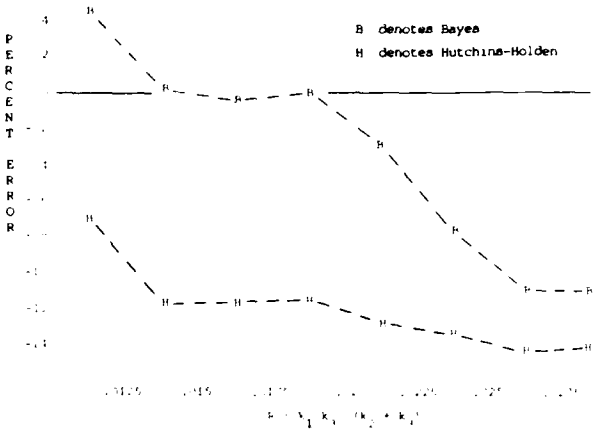
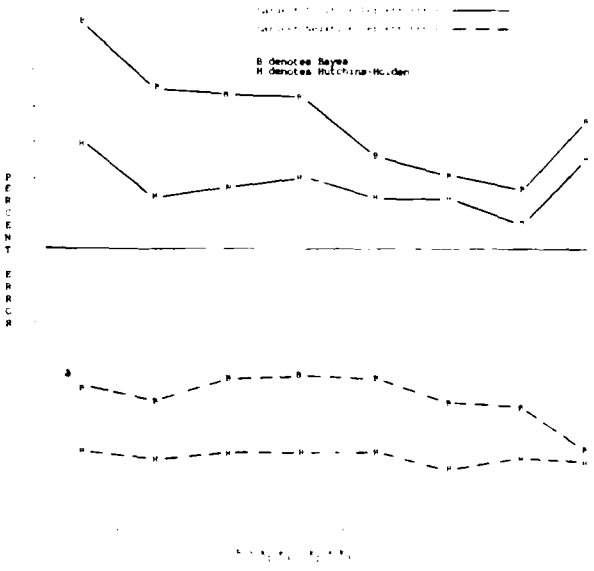


FIGURE 3. LARGEST POSITIVE AND NEGATIVE PERCENT ERROR



ESTIMATION OF DEATH DENSITY USING GROUPED CENSUS AND VITAL STATISTICS DATA

John J. Hsieh, University of Toronto

1. INTRODUCTION

This article develops a precise method for describing the distribution of the lifetime using census population and vital statistics data in cross-sectional studies. The description of the lifetime distribution will be made through estimation of survival function $p(x)$, conditional mortality probability function $q_x = 1 - p(x+1)/p(x)$, death density function $f(x)$, hazard function $h(x) = f(x)/p(x)$ and life expectancy (mean residual life function) $e(x) = \int_x^\infty p(y)dy/p(x)$.

To reduce random as well as systematic errors in the raw data employed in the estimation, population and death data are grouped according to the convention of abridged life tables in which the agespan $[0,\infty)$ is partitioned into $n = 19$ intervals $[x_i, x_{i+1})$, $i=0, 1, \ldots, 18$, with division points at $x_1 = 1$, $x_2 = 5$, $x_3 = 10$, ..., $x_{17} = 80$ and $x_{18} = 85$, so that the lengths of the age intervals are all five years except for the first two intervals which are one and four years, respectively, and for the last interval which is infinite. (For countries with high quality data the last division point may be taken to be $x_{19} = 90$ so that there are $n=20$ age intervals all told.) To estimate the functions for the lifetime distribution in a cross-sectional study we have to choose a base period and assume that the observed mortality schedule in the base period remains unchanged over time. To increase the reliability of the estimates, we shall use a base period of three calendar years.

Accuracy of estimation is achieved in each step by appropriate use of mathematical approximations, numerical quadrature and, in particular, spline interpolation, differentiation and integration. The available mortality data for the subdivisions of the first year of life and properties of the life time distribution allow an accurate determination of the two end conditions for the spline function. The use of spline methods serves to further smooth out errors arising from incomplete reporting and other sources that still remain in the data after age grouping.

The present method does more than just provide a new way of calculating the conventional life table functions (as in the construction of abridged life tables), it has several important features: (1). It allows one to calculate fundamental and useful functions of the lifetime distribution such as the death density function and the hazard function that are not available in a conventional life table, (2). it provides more accurate estimation of the conventional life table functions as well as other functions not found in published life tables than existing life table methods do, and, most importantly, (3). it allows one to compute these functions at any age point and for any age interval, in contrast to the traditional life table method which gives life table functions only at the age division points and for age intervals of a fixed mesh corresponding to the age grouping of the population and death data. (See Reed and Merrell 1939, Chiang 1968, Keyfitz and Frauenthal 1975.) Thus the present method not only allows the construction of complete as well as more refined life tables from abridged life tables but also expands the utility of the life table so constructed.

Section 2 describes the requisite data and calculation of death rates which are used in estimation of the lifetime distribution in Section 3. Methods for estimating the various functions depicting the lifetime distribution are described in Sections 3.1 through 3.6. Finally, an example illustrating the estimation of the five functions $f(x)$, $h(x)$, q_x, $p(x)$ and $e(x)$ using Canadian data is given in Section 4.

2. THE DATA AND CALCULATION OF DEATH RATES

The following data are required for the estimation of the above lifetime distribution functions according to the format of conventional abridged life tables using the proposed procedures: (1). Annual number of births B_j for each calendar year j of the 3-year base period plus one year preceding the base period, (2). deaths d_j in the last month of the first year of life for each calendar year j in the base period, (3). mid-year populations P_{ij} and deaths D_{ij} for the ith age group ($i=0, 1, \ldots, 18$) and jth calendar year ($j=1, 2, 3$), i.e., for the first year of life ($i=0$) and by 5-year age groups up to age 85 as well as those aged 85+, for each of the three calendar years in the base period, plus infant deaths for the year preceding the base period (D_{00}). These data are available from annual reports of vital statistics and mid-year population estimates or censuses published by most member nations of the United Nations and are also recorded in U.N. Demographic Yearbooks.

In order to estimate the lifetime distribution by the proposed procedures one has first to calculate the age-specific death rates M_i from the death and population data (D_{ij}, P_{ij}). The death rate M_i for the ith age group in the base period is defined as the total deaths in the ith age group over the base period A divided by the total observed person-years of exposure during the base period A for that age group. In symbols,

$$M_i = \frac{D_i}{\int_A P_i(t)dt}, \qquad \text{.....(1)}$$

where $D_i = \sum_{j=1}^{3} D_{ij}$ is the number of deaths observed during the 3-year base period for the ith age group and $P_i(t)$ is the population at time t for the ith age group. From formula (1) it is seen that the problem of calculating the death rate M_i reduces to the problem of numerically integrating out the person-year integral in the denominator of (1) in terms of the available mid-year population data. To this end, we take $P_i(t)$ to be a collocation polynomial (such as represented by Newton's forward formula) of order three interpolating to the three prescribed population data in the base period and perform the indicated integration to obtain:

$$M_i = D_i/Y_i, \quad \text{.....(2)}$$

where

$$Y_i = 3(3P_{i1}+2P_{i2}+3P_{i3})/8, \quad \text{.....(3)}$$

is the estimated person-years of exposure for the ith age group using the numerical quadrature described above.

Other methods of numerical quadrature could be used but the results would only differ insignificantly from that given by (3) (see Hsieh, 1978). In fact formula (3) is already considerably more accurate than the traditional method of calculating the death rate by dividing the average number of deaths in the base period by the mid-period population. The errors that still remain in the population data will be further smoothed by spline methods.

3. ESTIMATION OF THE LIFETIME DISTRIBUTION

In this section we shall show how the data and death rates described in section 2 are used to estimate various point and set functions representing the distribution of the lifetime. The procedures for the first year of life (i=0) and the last age interval (i=18) differ from those for the remaining age intervals. These two extreme age intervals require special treatments because mortality is extremely high and declines very sharply during the first year of life and because in the last open-ended age interval, the data tend to be thin and not of good quality.

The method starts with the estimation of conditional mortality probability q_i for the age intervals of the given mesh described in Section 1, using mathematical and numerical methods of approximations. This is followed by direct estimation and spline interpolation of the survival function at any age for the age segment [1,85). Estimates of the death density and hazard functions are obtained by spline differentiation and life expectancy by spline integration of the survival function. For ages beyond 85, we employ the Gompertz law of mortality to derive the estimate of these functions. Once the survival function at any age point is determined from spline (for $x \le 85$) or from Gompertz curve (for $x > 85$), it becomes a trivial matter to compute the conditional mortality probability and mean residence time for any age intervals.

3.1 Estimation of Conditional Mortality Probability q_i

The conditional mortality probability q_i is the probability of death in $[x_i, x_{i+1})$ given survival to age x_i, i=0, 1, ..., 18. They are estimated as follows:

1. For the first age interval (i=0)

$$q_0 = 1-(1-a)(1-b) \quad \text{.....(4)}$$

where

$$a = (1-f)D_0/B,$$
$$b = fD_0/(B'-(1-f)D_0'), \quad \text{.....(5)}$$

f is the separation factor, normally taken to be .11. $D_0 = \sum_{j=1}^{3} D_{0j}$ is the infant deaths in the base period, $D_0' = \sum_{j=0}^{2} D_{0j}$ is the infant deaths in the 3 calendar years starting from the year preceding the base period, $B = \sum_{j=1}^{3} B_j$ is the total births in the base period and $B' = \sum_{j=0}^{2} B_j$ is the total births in the 3 calendar years starting from the year preceding the base period. Formulas (4) and (5) are derived from probability arguments. Alternatively, q_0 may be obtained from construction of infant life tables (see Hsieh, 1985).

2. For the central age intervals (i=1, 2, ..., 17)

$$\ln(1-q_i) = -h_i(M_i+A_iB_i/Y_i), \quad \text{.....(6)}$$

where $h_i = x_{i+1}-x_i$ and M_i and Y_i are given by equations (2) and (3), respectively, and
where (a). for i=1,

$$A_1 = (725Y_1-418Y_2-162Y_3)/12825,$$
$$B_1 = (-1120M_1+1444M_2-324M_3)/855; \quad \text{.....(7)}$$

(b). for i=2, ..., 15,

$$A_i = (9Y_{i-1}-3Y_i-5Y_{i+1}-Y_{i+2})/192,$$
$$B_i = (-3M_{i-1}+3M_i+7M_{i+1}-M_{i+2})/8; \quad \text{.....(8)}$$

(c). for i=16 and 17,

$$A_i = (Y_{i-2}+2Y_{i-1}-3Y_i)/48,$$
$$B_i = (M_{i-2}-4M_{i-1}+3M_i)/2; \quad \text{.....(9)}$$

The derivation of formulas (6) - (9), which employs solution of an integral equation using Taylor expansion and Newton's formulas, is given in Hsieh (1988).

3. For the last age interval (i=18)

$q_{18} = 1$, since everybody has to die eventually.

Estimation of conditional mortality probability for any age intervals other than $[x_i, x_{i+1})$, i=0, 1, ..., 18, is given in section 3.6.

3.2 Estimation of Survival Function

Unlike the conditional mortality probability q_i discussed in Section 3.1 and the conditional mean residence time and mortality probability to be discussed in Section 3.6 which are set functions, the remaining four functions to be studied in Section 3.2-3.5 are all point functions.

The survival function p(x) is the probability of surviving to age x. The survival function $p(x_i)$ at the division points x_i is obtained directly from the conditional mortality probabilities q_i as follows: For i=1, 2, ..., 18,

$$p(x_i) = \prod_{j=0}^{i-1}(1-q_j). \quad \text{.....(10)}$$

The proof of (10) is straight forward. Clearly, at the two ends of the agespan, $p(0)=1$ and $p(\infty)=0$.

To estimate the survival function at ages other than the division points, i.e., at $x \neq x_i$, i=1, 2, ..., 18, different procedures are required for each of the three main age segments. For ages under one year, the estimation method is given in Hsieh (1985). For ages from 1 to 85 years the method of spline interpolation will be used. To this end, we pass an interpolating spline $s_p(x)$ through the prescribed $p(x_i)$ values, i=1, 2, ... 18, and take this spline function as an estimate of the survival function p(x), for all $x \in [1,85)$. From our knowledge of the pattern of the survival curve for the human lifetime and the availability of the mortality data in cross-sectional studies, the complete interpolating cubic spline would be the best choice among all polynomial spline functions possessing optimum approximation properties.

An interpolating cubic spline can be represented mathematically in a number of ways (see, e.g., Ahlberg, Nilson and Walsh 1967, de Boor 1978 and Schumaker 1981). We shall choose the following representation for the cubic spline interpolating to the prescribed data $p_i \equiv p(x_i)$, i=1, 2, ..., 18: For $x \in [x_i, x_{i+1})$,

$$s_p(x) = m_i \frac{(x_{i+1}-x)^2(x-x_i)}{h_i^2} - m_{i+1} \frac{(x-x_i)^2(x_{i+1}-x)}{h_i^2} + p_i \frac{(x_{i+1}-x)^2[2(x-x_i)+h_i]}{h_i^3} + p_{i+1} \frac{(x-x_i)^2[2(x_{i+1}-x)+h_i]}{h_i^3}, \quad \text{.....(11)}$$

For a given x in $[x_i, x_{i+1})$, i=1, 2, ..., 17, all quantities in (11) are known except the slopes $m_i = s'_p(x_i)$ at the division points. These parameters are determined by solving the following system of 16 linear equations.

$$h_i m_{i-1} + 2(h_i+h_{i-1})m_i + h_{i-1}m_{i+1} = 3\left[\frac{h_{i-1}}{h_i}(p_{i+1}-p_i) + \frac{h_i}{h_{i-1}}(p_i-p_{i-1})\right], \quad \text{.....(12)}$$

for i=2, 3, ..., 17, with the two boundary conditions:
(a) the first endslope

$$m_1 = -(365/31)p_1 d/(B-D_0-d), \quad \text{.....(13)}$$

where $d = \sum_{j=1}^{3} d_j$.

(b) the last endslope

$$m_{18} = -p_{18} M_{17}^{\frac{3}{2}} / M_{16}^{\frac{1}{2}}. \quad \text{.....(14)}$$

Equation (12) is derived from equation (11) by differentiating twice with respect to x and using the continuity constraints of cubic splines at the interior division points. The two end (boundary) conditions (13) and (14), which estimate the slopes of the survival function at the two boundaries, are accurately determined from properties of the lifetime distribution. The tridiagonal form of the coefficient matrices of (12) allows the linear systems to be easily solved using a computer by Gaussian elimination with partial pivoting. Furthermore, the diagonal dominance and symmetric characteristic of the matrix guarantees stable results with minimum accumulation of rounding errors. Once the parameters m_i are solved for from (12) with boundary conditions (13) and (14), the survival function at any age x can be calculated from (11). Note that, as expected, when $x=x_i$ or $x=x_{i+1}$, equation (11) reduces to $s_p(x_i) = p_i$ or $s_p(x_{i+1}) = p_{i+1}$, respectively.

For the last open-ended age interval (i.e. for ages beyond $x_{18} = 85$), we discard the data in this age group for lack of reliability and adopt the Gompertz law of mortality. By fitting the Gompertz survival curve to the three prescribed survival functions at ages $x_{16}=75$, $x_{17}=80$, and $x_{18}=85$, namely, p_{16}, p_{17} and p_{18}, we obtain for $t=x-x_{18} \geq 0$,

$$p(x) = p_{18}\, g^{(c^t-1)c^{10}} \quad \text{.....(15)}$$

$$\text{where } c = \left[\frac{\ln(p_{18}/p_{17})}{\ln(p_{17}/p_{16})}\right]^{\frac{1}{5}}. \quad \text{.....(16)}$$

$$\ln g = \frac{\ln(p_{17}/p_{16})}{c^5 - 1} \quad \text{.....(17)}$$

and p_{16}, p_{17} and p_{18} are given by (10).

3.3 Estimation of Death Density Function

The death density function f(x) is the probability per unit time of dying in the instant immediately following age x. For the age segment [1,85], the death density function at a division point x_i, i=1, 2, ..., 18 is simply the negative of the slope of the survival function at that point. The spline estimate of the death density at ages other than the division points are obtained by differentiating the spline estimate of the survival function (11) to yield:

$$\begin{aligned} s_f(x) &= -s'_p(x) \\ &= -h_i^{-3}[h_i(x_{i+1}-x)(2x_i+x_{i+1}-3x)m_i \\ &\quad -h_i(x-x_i)(2x_{i+1}+x_i-3x)m_{i+1} \\ &\quad +6(x_{i+1}-x)(x-x_i)(p_{i+1}-p_i)]. \end{aligned} \quad \text{.....(18)}$$

Finally the spline parameters m_i obtained in Section 3.2 are substituted into (18) to determine the estimate $s_f(x)$ of the death density function f(x). Note that when $x=x_i$, equation (18) reduces to

$$s_f(x_i) = -m_i, \quad \text{.....(19)}$$

as pointed out at the beginning of this subsection.

For ages beyond $x_{18}=85$, the death density is estimated by differentiating (15) with respect to t to get, for $t=x-x_{18} \geq 0$,

$$f(x) = -p_{18}(\ln g)(\ln c)c^{t+10}g^{(c^t-1)c^{10}}, \quad \text{.....(20)}$$

where p_{18}, c and g are given by (10), (16) and (17), respectively.

For ages under one year the estimation method is given in Hsieh (1985).

3.4 Estimation of Hazard Function

The hazard function h(x) is the conditional

probability per unit time of dying in the instant immediately following age x given survival to age x. Once the death density function and survival function are determined as described in Sections 3.2 and 3.3 above, the hazard function is taken as their ratio. For x in [1,85), we divide (18) by (11) to get a spline estimate of the hazard function:

$$s_h(x) = s_f(x)/s_p(x), \qquad \text{.....(21)}$$

Note that when $x=x_i$, equation (21) reduces to

$$s_h(x_i) = -m_i/p_i, \qquad \text{.....(22)}$$

in view of (19) and the comments at the end of the paragraph following equation (14).

For ages beyond $x_{18}=85$, we divide (20) by (15) to get, for $t=x-x_{18}\geq 0$,

$$h(x) = -(\ln g)(\ln c)c^{t+10}, \qquad \text{.....(23)}$$

where c and g are given by (16) and (17), respectively.

For ages under one year the estimation method is given in Hsieh (1985).

Since differentiation of a spline results in a spline (of one lower order) which still possesses optimum approximation properties, the use of spline method of differentiation, unlike other numerical differentiation procedures, greatly enhance the accuracy of the estimates of both death density and hazard functions. Before the advent of spline functions, the difficulty with numerical differentiation has been the main reason why conventional life tables do not include death density and force of mortality.

3.5 Estimation of Life Expectancy

The life expectancy at age x, e(x), is the average remaining lifetime for a person alive at age x. While estimation of density and hazard functions requires differentiating the survival function, estimation of life expectancy requires integrating the survival function. (Note that both integration and differentiation of splines result in splines.) For the first year of life, we use the mean value theorem of integral calculus to obtain an estimate of the person-year integral

$$L_0 \equiv \int_0^1 p(x)dx = 1-(1-f)q_0, \qquad \text{.....(24)}$$

where f is the separation factor defined in (5). For ages beyond $x_{18}=85$, we integrate (15) from x ($\geq x_{18}$) to ∞ to get the person-years lived beyond age x.

$$T(x) \equiv \int_x^\infty p(t)dt = p_{18}\, g^{-c^{10}}E_1(-c^{x-x_{18}}\ln g)/\ln c, \qquad \text{.....(25)}$$

where the exponential integral $E_1(k)=\int_k^\infty (e^{-u}/u)du = -\gamma - \ln k - \sum_{n=1}^{\infty}(-1)^n\frac{k^n}{n\,n!}$ (with $\gamma = 0.5772156649...$ being Euler's constant). When $x = x_{18}$, (25) becomes

$$T(x_{18}) \equiv \int_{x_{18}}^\infty p(t)dt = p_{18}\, g^{-c^{10}}E_1(-c^{10}\ln g)/\ln c, \qquad \text{.....(26)}$$

For x in $[x_i,x_{i+1})$, i=1, 2, ..., 17, we have the person-years lived beyond age x:

$$T(x) = \int_x^\infty p(t)dt = \int_x^{x_{i+1}} s_p(t)dt + \int_{x_{i+1}}^{x_{18}} s_p(t)dt + \int_{x_{18}}^\infty p(t)dt = I(x) + \sum_{j=i+1}^{17} L_j + T(x_{18}), \qquad \text{.....(27)}$$

where

$$L_i = \int_{x_i}^{x_{i+1}} s_p(t)dt = h_i(p_i+p_{i+1})/2 + h_{i+1}^2(m_i-m_{i+1})/12. \qquad \text{.....(28)}$$

is obtained by integrating (11) from x_i to x_{i+1} and

$$I(x) = \int_x^{x_{i+1}} s_p(t)dt = (x_{i+1}-x)(p(x)-p_{i+1})/2-(x_{i+1}-x)^3(p''(x)+p''(x_{i+1}))/24, \qquad \text{...(29)}$$

with

$$p''(x) = p''(x_i)(x_{i+1}-x)/h_i - p''(x_{i+1})(x_i-x)/h_i \qquad \text{.....(30)}$$

and

$$p''(x_i) = -2h_i^{-2}[(2m_i+m_{i+1})h_i+3(p_i-p_{i+1})]. \qquad \text{.....(31)}$$

$$p''(x_{i+1}) = 2h_i^{-2}[(m_i+2m_{i+1})h_i+3(p_i-p_{i+1})]. \qquad \text{.....(32)}$$

Note that for i=1, 2, ..., 17, $I(x_i) = L_i$ and $I(x_{i+1})=0$ so that at the division points,

$$T(x_i) = \sum_{j=i}^{17} L_j + T(x_{18}) \qquad \text{.....(33)}$$

and that

$$T(0) = \sum_{j=0}^{17} L_j + T(x_{18}). \qquad \text{.....(34)}$$

With tail person-year integral T(x) computed as above, the life expectancy (mean residual life function) is estimated by

$$e(x) = T(x)/p(x), \qquad \text{.....(35)}$$

where for $x\in[1,85)$, $p(x)=s_p(x)$ and T(x) are given by (11) and (27), respectively, and for $x\geq 85$, p(x) and T(x) are given by (15) and (25), respectively. Note that at the division points $x=x_i$, (35) reduces to

$$e(x_i) = (\sum_{j=i}^{17} L_j + T(x_{18}))/p_i. \qquad \text{.....(36)}$$

3.6 Estimation of Conditional Mortality Probability and Conditional Mean Residence Time (General)

For $0\leq x\leq y<z<\infty$, the conditional probability of death in [y,z] given survival to age x, denoted by q[x;y,z], is the ratio of the difference between the survival function at y and the survival function at z to the survival function at x. If $x,y,z\in[1,85)$, then this conditional mortality probability is estimated by

$$q[x;y,z] = [s_p(y)-s_p(z)]/s_p(x), \qquad \text{.....(37)}$$

where the spline survival functions $s_p(.)$ are obtained from (11). If either z alone, or y and z, or x,y and z are greater than 85, then the corresponding

TABLE 1 Estimation of f(x), h(x), q_x, p_x and e(x) for Canadian Males, 1980-82.

(1) Beginning age of interval x	(2) Death density function $10^3 f(x)$	(3) Hazard function $10^3 h(x)$	(4) Conditional Mortality Probability $10^3 q_x$	(5) Survival function $10^3 p(x)$	(6) Life Expectancy e(x)
0	18008.79*	18008.79*	10.91	1000.00	71.87
1	1.53	1.55	0.85	989.09	71.66
2	0.81	0.82	0.64	988.25	70.73
3	0.49	0.50	0.45	987.62	69.77
4	0.33	0.33	0.42	987.17	68.80
5	0.31	0.31	0.35	986.76	67.84
6	0.35	0.35	0.33	986.41	66.86
7	0.36	0.37	0.32	986.09	65.88
8	0.35	0.36	0.31	985.77	64.90
9	0.33	0.33	0.30	985.47	63.91
10	0.27	0.27	0.25	985.17	62.93
11	0.23	0.23	0.24	984.92	61.95
12	0.27	0.27	0.31	984.69	60.96
13	0.36	0.37	0.45	984.38	59.98
14	0.54	0.55	0.66	983.94	59.01
15	0.78	0.79	0.93	983.29	58.05
16	1.03	1.05	1.16	982.38	57.10
17	1.24	1.26	1.34	981.24	56.17
18	1.38	1.41	1.47	979.92	55.24
19	1.49	1.52	1.55	978.48	54.32
20	1.53	1.57	1.58	976.96	53.41
21	1.55	1.59	1.59	975.42	52.49
22	1.55	1.59	1.59	973.87	51.57
23	1.54	1.58	1.56	972.32	50.65
24	1.50	1.54	1.52	970.80	49.73
25	1.44	1.49	1.46	969.33	48.81
26	1.39	1.44	1.41	967.91	47.88
27	1.34	1.39	1.37	966.55	46.95
28	1.31	1.3	1.34	965.22	46.01
29	1.28	1.33	1.32	963.93	45.07
30	1.27	1.32	1.32	962.66	44.13
31	1.27	1.32	1.33	961.39	43.19
32	1.29	1.34	1.35	960.11	42.24
33	1.31	1.37	1.40	958.81	41.30
34	1.37	1.43	1.46	957.47	40.36
35	1.42	1.49	1.53	956.07	39.42
36	1.51	1.58	1.63	954.61	38.48
37	1.61	1.69	1.75	953.05	37.54
38	1.73	1.82	1.89	951.39	36.60
39	1.87	1.97	2.05	949.59	35.67
40	2.03	2.14	2.23	947.64	34.74
41	2.21	2.34	2.45	945.53	33.82
42	2.43	2.58	2.71	943.21	32.90
43	2.69	2.86	3.01	940.65	31.99
44	2.98	3.18	3.35	937.82	31.08
45	3.31	3.54	3.73	934.68	30.19
46	3.67	3.94	4.16	931.20	29.30
47	4.07	4.39	4.62	927.32	28.41
48	4.50	4.88	5.13	923.04	27.55
49	4.97	5.41	5.68	918.30	26.69
50	5.47	5.99	6.28	913.09	25.83
51	5.72	6.30	6.95	907.35	24.98
52	6.60	7.33	7.69	901.05	24.15
53	7.25	8.11	8.49	894.12	23.25
54	7.94	8.96	9.38	886.53	22.54
55	8.69	9.90	10.34	878.21	21.76
56	9.47	10.90	11.37	869.13	20.98
57	10.29	11.98	12.48	859.25	20.21
58	11.15	13.14	13.66	848.52	19.46
59	12.04	14.38	14.93	836.93	18.72
60	12.95	15.71	16.29	824.44	18.00
61	13.93	17.17	17.80	811.01	17.29
62	14.96	18.78	19.47	796.57	16.60
63	16.07	20.57	21.32	781.06	15.96
64	17.25	22.56	23.37	764.41	15.25
65	18.49	24.77	25.62	746.55	14.60
66	19.76	27.16	28.01	727.42	13.97
67	20.99	29.69	30.55	707.04	13.36
68	22.20	32.39	33.25	685.44	12.76
69	23.38	35.29	36.16	662.65	12.19
70	24.54	38.42	39.29	638.69	11.63
71	25.65	41.80	42.65	613.60	11.08
72	26.69	45.44	46.28	587.43	10.55
73	27.68	49.40	50.24	560.24	10.04
74	28.60	53.75	54.57	532.09	9.55
75	29.46	58.57	59.33	503.06	9.07
76	30.21	63.84	64.47	473.21	8.61
77	30.79	69.54	70.03	442.70	8.17
78	31.19	75.77	76.09	411.70	7.74
79	31.43	82.63	82.77	380.37	7.34
80	31.50	90.28	90.11	348.89	6.96
81	31.29	98.57	97.77	317.45	6.60
82	30.70	107.18	105.60	286.42	6.26
83	29.72	116.02	113.46	256.17	5.94
84	28.36	124.87	121.06	227.10	5.64
85	27.33	136.93	133.14	199.61	5.01
86	25.78	149.01	144.00	173.03	4.71
87	24.02	162.15	155.66	148.12	4.41
88	22.07	176.44	168.16	125.06	4.14
89	19.97	192.00	181.56	104.03	3.87
90	17.79	208.94	195.89	85.14	3.62
91	15.57	227.36	211.20	68.46	3.39
92	13.36	247.41	227.53	54.00	3.16
93	11.23	269.23	244.92	41.72	2.95
94	9.23	292.97	263.40	31.50	2.75
95	7.40	318.81	282.99	23.20	2.57
96	5.77	346.92	303.72	16.64	2.39
97	4.37	377.51	325.60	11.58	2.22
98	3.21	410.80	348.62	7.81	2.07
99	2.27	447.03	372.78	5.09	1.92
100	1.55	486.45	398.06	3.19	1.79
101	1.02	529.35	424.41	1.92	1.66
102	0.64	576.03	1000.00	1.11	1.54

* Taken from Hsieh (1985)

spline survival functions $s_p(.)$ in (37) are to be replaced by the Gompertz survival functions p(.) given by (15). When z=x+1 and y=x, then (37) reduces to the complete life table function

$$q_x = 1 - p(x+1)/p(x). \quad \text{.....(38)}$$

Furthermore, when $z=x_{i+1}$ and $x=y=x_i$, then (37) reduces to the abridged life table function of Section 3.1:

$$q_i = 1 - p_{i+1}/p_i. \quad \text{.....(39)}$$

For $0 \le x \le y < z < \infty$, the conditional mean lifetime, e[x;y,z], represents the average number of years lived in [y,z] for a person alive at age x. It is the conditional expected life in [y,z] given survival to age x and can be expressed as the difference between the person-years lived beyond y and the person-years lived beyond z divided by the survival function at age x. In symbols,

$$e[x;y,z] = [T(y)-T(z)]/p(x), \quad \text{.....(40)}$$

To estimate e[x;y,z], we substitute in (40) T(x) and p(x) given by (27) and (11), respectively, if $x \in [1,85)$, and by (25) and (15), respectively, if x>85. If y=x, then (40) reduces to the well-known Markov mean residence time. When y=x and $z=\infty$, then (40) reduces to the usual life expectancy e(x) of Section 3.5.

The general conditional mortality probabilities and mean life times described in this subsection would provide additional instruments for mortality analysis.

4. AN EXAMPLE

We have applied the methods of estimation described in Sections 3.1 through 3.6 to the data specified in Section 2 for Canadian males. These data are available from the Annual Vital Statistics of Canada (1979-1982). We use a base period of three years from 1980 to 1982, inclusive, to calculate the age-specific death rates M_i and person-years of exposure Y_i from equations (2) and (3), which, in turn, are used to compute q_i from equations (6)-(9), i=1, 2, ..., 17. The value for q_0 has been computed from equations (4) and (5). The spline methods and Gompertz model of Sections 3.2 to 3.6 were then employed to estimate f(x), h(x), q_x, p(x) and e(x) for age x from the q_i values so computed. Table 1 shows the final results for these five functions at x=0, 1, 2, ..., 101, 102. Computation of these functions at other age points can be done similarly.

Acknowledgements

This research was supported under the Natural Sciences and Engineering Research Council of Canada Operating Grant A9253

REFERENCES

Ahlberg, J.H., Nilson, E.H., and Walsh, J.L. (1967): The Theory of Splines and their Applications. Academic Press, New York.

de Boor, C. (1978): A Practical Guide To Splines Springer-Verlag, New York.

Chiang, C.L. (1968): Introduction to Stochastic Processes in Biostatistics. Wiley, New York.

Hsieh, J.J. (1978): Computation of Person Years in the Construction of Abridged Life Tables. Proc. Am. Statist. Assoc., Soc. Statist. Sec., 476-481.

Hsieh, J.J. (1985): Construction of Expanded Infant Life Tables: A Method Based on a New Mortality Law. Math. Biosci. 76, 221-242.

Hsieh, J.J.: Life Table Analysis. Academic Press, New York. (In press)

Keyfitz, N. and Frauenthal, J. (1975): An Improved Life Table Method. Biometrics 31: 889-899.

Reed, L.J. and Merrel, M. (1939): A Short Method for Constructing An Abridged Life Table. Am. J. Hygiene 30: 33-62.

Shumaker, L.L. (1981): Spline Functions. Wiley, New York.

Statistics Canada (1979-1982): Vital Statistics, Catalogue 84-206, Vital Statistics and Disease Registries Section, Ottawa.

Extracting Records from New Jersey's Multiple Cause of Death Files

Giles Crane, New Jersey Department of Health

Abstract

A simple microcomputer system has been developed using off-the-shelf components which permits local access in an acceptable time frame to seven years of New Jersey multiple cause of death data assembled and distributed by the National Center for Health Statistics. The system includes hardware and software, and illustrates a trade-off between speed and specificity of access to approximately 70,000 records per calender year. Applications to the epidemiology of drowning and sickle cell anaemia will be discussed with timing information and rules of thumb for similar investigations. The numbers of causes per person in New Jersey will be summarized in several tables. If time permits, the further analysis of abstracts from this data will be illustrated by three short examples: conventional statistial analysis, a computationally intensive method, and an application of artifical intelligence tecnique.

1. Introduction

New Jersey is the fifth smallest state in the Union, with a population of approximately 8 million people, approximately 100,000 births per year, and about 70,000 deaths per year. As part of national and international public health efforts, causes of death are coded in the ICD9 (International Classification of Disease Codes) (0) and entered on the death certificate. NJ at present prepares computer readable, restricted access tapes of death certificates which include the "underlying cause" of death", but as yet NJ does not prepare multiple cause of death tapes.

In the view of some epidemiologists, the multiple cause of death tapes are the single, most important source of epidemiological information for health research. It was viewed as imperative to improve access to these requests by medical researchers. Some required over 6 months or, in several instances, were never completed due to the press of operational processing and maintenance requirements at the Department minicomputer unit.

There are many other NJ health outcome databases which offer information for research and yet which are in need of improved access:

1. Birth Tapes
2. Death Tapes
3. Hospital Discharges
4. Drug Treatment Discharges
5. Cancer Registry
6. Birth Defects Registry
7. Medical Claims Tapes
8. AIDS Registry
9. Fetal Death Tapes
10. Poisoning Reports
11. Cervical Cancer Screening Reports
12. Communicable Diseases
13. Family Planning
14. Hemophilia Financial Assistance
15. Mental Retardation Services
16. Fatal Accidents Reports
17. Homicide Reports
18. Community Mental Health Centers

2. Multiple Cause of Death Records (1,2)

The distributing part of the path of the multiple cause of death information for this project was composed of the following steps: (from the top down--there also is a bottom up set of steps)Federal Vital statistics--coding PRIME minicomputer or Ed.Comp. Network -- reblocking. NJ Center for Health Statistics -- minitape. Micro-computer -- stripping, packing, compacting, and access.

File sizes for the 7 years of Multiple Cause of Death records are:

YEAR	RECORDS	MBYTES
1979	69,969	30.79
1980	71,202	31.33
1981	69,557	30.61
1982	69,854	30.74
1983	71,627	31.53
1984	71,743	31.57
1985	73,520	32.34

Under the present system, the 1986 records may become available sometime in September 1988. There appears to be scope for improvement by new technology, or organization, or both.

The MCD records were stripped of unnecessary or redundant fields, including "record access" fields, and only 10 of a possible 20 "entity: axis" (original ICD9 codes) were preserved. These short MCD records were packed, two characters (0123456789 Z) per byte, and then subjected to a compacting utility. The records sizes are shown below:

440 bytes M.C.D. record (442 with CR,LF)
129 Short M.C.D. record (no CR,LF)
65 Packed, short M.C.D. record (no CR,LF)

A C program, called GETMCD, written to strip the records, pack them, and also to upack and access them by ICD9 codes. Principal access is by Multiple Cause of death codes (ICD9CM):

Underlying cause code--4 characters e.g. 0460, 046. Contributing cause entity axis codes-7 character.

char 1 Death Cert. Line no. 1..6
2 Sequence No. on line 1..7
3-6 Cause code: 4 char ICD9CM
7 1 if nature of cause, 0 otherwise

Causes are specified somewhat like MS-DOS file names, in which "?" denotes any single character and "_" is translated into a blank. For example, sickle-cell disease is called for by "??28260", i.e. any death certificate line, any code on line, ICD9 code 282.6, and not a nature of cause code. Lung disease is accessed by several codes: "??500?0", " ??501?0", "??502?0", "??503?0", "??504?0", "??505?0".

3. Hardware (3)

The hardware used in this realization of the access system consisted of a Compaq Deskpro dual speed microcomputer (640K memory, 30 Mbyte hard

disk, 360K floppy disk, Irwin 10 Mbyte DC1000 minicartiridge tape drive).

4. Software (3,4)
The software consisted of Compaq MD-DOS Version 3.00 as an operating system, the TAPE.EXE Version 1.05 tape utility, Turbo C compiler, a public domain packing program, a C program for selection at high speed, a simple editor, and several MS-DOS command files.
Detailed processing of the data abstracted from the files is done by a range of techniques from printing the small file and inspecting the data to more elaborate database and statistical packages favored by individual investigators (LOTUS, DBASE, SPSS-X GLIM3, PRODAS).

5. Problems encountered
Typical tape and file handling problems were encountered such as a tape with several bad blocks, hardware problem with the reel magnetic tape uit, inadvertant changes in directory names when processing different years, and inconsistencies in using CR,LF as record terminators.

6. Tests of the system
The C program GETMCD was tested in several ways. Results of various test requests were checked against the NCHS Underlying Cause counts and against control tables in the MCD tpae documentation (1).
The C debugging process also provided further assurrance that the program was functioning as desired.
The packed files were inspected by an independent software package which displayed the packed records in HEX format, a visual unpacking of the 2 "nibbles: which make up an 8-bit byte.
The entire process was run forward and backward, selecting all records.
Finally, a detailed investigation of the full records from 1984 provided additional checks of the packing and selecting process. (See next section.)

7. 1984 Multiple Cause of Death Records
The number of causes of death for 1984 in New Jersey were analysed briefly in order to count the records with over 10 contributing causes of death, to provide further checks of the system, and to provide general guidance for other researchers. Quite probably, this is the first time these figures have been published.

Number of Contributing Causes	Number of People	Percent
1	9,487	13
2	17,519	24
3	19,262	27
4	13,440	19
5	6,883	10
6	3,028	4
7	1,250	2
8	511	1
9	205	0.3
10	92	0.1
11-14	66	0.09

The average number of contributing causes for 1984 was 3.14. (It may be noted that the Certificate of Death has three lines for contributing cause of death.) The average number of causes was calculated by age, race, sex, marital status, and presense of autopsy:

Age		Race	Sex	
0	2.97	Other Asian 2.89		
1-4	3.30	White 3.14	male	3.11
5-9	3.44	Black 3.15	female	3.16
10-14	3.30	Am. Indian 3.12		
15-24	3.49	Chinese 2.93		
25-34	3.22	Japanese 2.86		
35-44	3.00	Hawaiian 2.5 (2 only)		
45-54	2.91	Fikipino 3.36		
55-64	2.93	All other 2.91		
65-74	3.10			
75-84	3.25			
85+	3.25			

Marital Status		Autopsy	
single	3.14	yes	3.35
married	3.08	no	3.13
widowed	3.21		
divorced	3.03		
not stated	2.91		

8. Applications
In the 5 months of operation, this system has been applied by members of the Office of Research, the Divisions of Maternal and Child Health, the Occupational Health Program, Narcotic and Drug Abuse, Cardiovascular Disease Unit and the AIDS Division, as well as by other members of the Office of the Commissioner of Health.

In an investigation of drowning and near drownings (5), immersion injuries leading to death in New Jersey were identified and selected for 1981-85. These records were matched with hospital discharge data and further analysis was done in order to calculate incidence rates and case fatality ration by age, race, sex, and county.
An investigation of sickle cell disease (6), records were selected using ICD9CM codes for 282.4 (Thalassemias), 282.6 (Sickle cell anemia), and 282.7 (Other hemoglobinopathies), and a short paper was published giving the results and implications.
After an initial study of hospital costs (7), several AIDS related studies are underway which involve data from three other sources in addition to selections from the Multiple Cause of Death Records.

9. Examples of further analysis methods.
The applications discussed were developed using conventional statistical analysis techniques (8) including cross tabulations, regression, histogram and bar charts, and confidence limits. More computationally intensive methods such as the bootstrap and jacknife (9) are being applied

using short C programs and detailed applications of elaborate statistical packages (8).
At this time artificial intelligence techniques (10) are the subject of experiment, and various strategies for employment of the AI inference engine are being considere. One possible project involving the learning aspect of AI would be to build a knowledge base of related diagnoses.

10. Further work
The system could be improved by purchase of a faster sequential, access device (faster minitape, CD Rom, video cassette), or dedication of the hard disk to storage of the files. The C program could be further optimized for speed, either applying an optimizing C compiler (for example, Microsoft) or re-writing the C program. The use of the file compaction program might be eliminated, or compaction could be made record selection a part of the compaction utility.) Combining the compaction, selection, and tape reading utility into one program would reduce the passes through the tape from 3 to 1.
Alternate storage orders for the records might also be considered, perhaps by ICD9-CM code over all years. The file-by-file rather than record-by-record capability of the current tape drive is a constraint on the problem, which has been considered many times for record-by-record tape systems.
Barriers to improvement are increased cost of hardware, other users competing for time on the microcomputer, manufacturers who do not wish to release software to directly access minitape drives.
Arranging the tape files in the best order by learning the years most frequently requested did not increase efficiency here since the minitape cartridge rewound automatically after restoring a single file. It was decided to place the yearly files in sequential order, 4 to a tape. Having the last few years on the last tape will eliminate some tape changes. A 40 Mbyte tape drive will probably be adequate to hold all MCD data for the life of the system (see below, Further Work).

11. Summary
Before this access system was devised, requests for abstracts from MCD files from researchers at the NJ Dept. of Health could require months or were not possible. After the MCD files became locally available, more than 10 different researchers were able to access this data, usually within two days. Studies involving this data have been presented at meetings of the local working group on health data, a national conference, and an international conference.
Pre-processing of the tape information (transference to microcomputer, stripping of redundant field, packing, compacting, transfering to minitape) can be viewed as moving the common part of the time required for satisfying any request to time required to prepare the system. This appears to be a general principal which motivates the formation of many systems.

12. References
(0) The International Classification of Diseases 9th Revision Clinical Modification, Second Edition, Sept. 1980. U.S. Department of Health and Human Services, Public Health Services, Health Care Financing Administration.
(1) Public Use Data Tape Documentation: Multiple Cause of Death for ICD9-9 1985 Data. U.S. Department of Health and Human services, Public Health service, National Center for Health Statistics (Hyattsville, Maryland, August 1987).
(2) MCD Short Record Documentation. New Jersey State Department of Health Division of Research, Policy, and Planning, Office of Research. (Trenton, New Jersey, June 1988).
(3) COMPAQ Computer Corporation (1983). Houston, Texas 77070.
(4) Irwin Magnetics (1983). 2101 Commonwealth Blvd., Ann Arbor, Michigan 48105.
(5) Daniel Fife, M.D., Sharon Scipio, and Giles Crane (1988). Fatal and Non-Fatal Immersion Injuries Among New Jersey Residents. New Jersey Dept. of Health.
(6) E. Rappaport, M.D., Gile Crane, and Daniel Fife, M.D. (1988) Hospitalizations of Hemoglobinopathy Patients in New Jersey. (Accepted by the American Public Health Association Annual Meeting).
(7) Statistical computing work for: Molly Joel Coye, Richard Conviser, Howard S. Berliner, Christine M. Grant (1988). RESULTS OF A STATEWIDE STRATEGY TO CONTAIN HOSPITAL COSTS OF AIDS PATIENTS. Prepared for the First International Conference on AIDS, Stockholm, Sweden, June 1988.
(8) PRODAS Programming Language. Conceptual Software Systems, Houston Texas.
(9) Bradley Efron (1982). The Jackknife, the Bootstrap, and Other Resampling Plans. CBMS-NSF Regional Conference series in Applied Mathematics Number 38. Society for Industrial and Applied Mathematics (Philadelphia, Pennsylvania).
(10) W.D. Burnham and A.R. Hall (1985). Prolog Programming and Applications. (John Wiley & Sons, New York).
Appendix: Estimate of time and cost to change birth or death certificate.
At the presentation of this paper, one conference member asked how long and how much it would cost to correct a birth or death certificate in New Jersey. Corrections to birth certificates are made at no cost and the correction from is inserted at the end of the queue of certificates on hand. At this date, February birth certificates are being entered and so there will be a delay of 6 months. Also, the usual charge will be made for a new certificate. As for corrections to death certificates, a similar policy holds, but a law state that new death certificates must be

available with 60 days. Experience here indicates that a correction which is passed down from Federal Vital Statistics and which requires verification at the office nearest the site of death may require as long as 6 months.

Acknowlegments

The author thanks the staff of the Center for Health Statistics and the Office of Research for their encouragement and help.

XIX. IMAGE PROCESSING

A PROBABILISTIC APPROACH TO RANGE DATA SEGMENTATION

EZZET AL-HUJAZI, WAYNE STATE UNIVERSITY
ARUN SOOD, GEORGE MASON UNIVERSITY

ABSTRACT

In this paper we present a region growing approach for segmenting range images based on the H (Mean Curvature) and K (Gaussian Curvature) parameters. Range images are unique in that they directly approximate the physical surfaces of a real world 3-D scene. H and K are defined from the fundamental theorems of differential geometry, and provide visible, invariant pixel labels that can be used to characterize the scene. The sign of H and K can be used to classify each pixel into one of eight possible surface types. Due to the sensitivity of these curvature parameters to noise, the computed HK-sign map does not directly identify surfaces in the range image. In this paper a probabilistic approach for the segmentation of the range image is suggested. The image is modeled as a Markov Random Field on a finite lattice. The prior knowledge about the solution is expressed in the form of a Gibbs probability distribution. This approach allows the integration of the output of a number of modules in an efficient way. The performance of the proposed technique on a number of range images will be presented.

1. INTRODUCTION

The statistical techniques for modeling and processing image data has seen an increasing interest in computer vision literature resently. Most of the work has been directed toward application of Markov Random Field (MRF) models to problems in texture modeling and classification and problems in segmentation and restoration of noisy and textured images [2,4,5,6,7,9].

In differential geometry the information given by the sign of H and K can be used to classify a surface point into one of eight possible labels. These two surface curvatures are derived from the first and second fundamental forms. They are sensitive to noise and the resulting HK-sign map does not correspond directly to surfaces in the image and thus it has to be further processed.

In this paper an algorithm based on MRF and edge models is suggested for processing the HK-sign map. This approach is chosen because it allows an analytical basis for integrating a number of object features. A variable neighborhood area is used for the MRF which gives a good compromise between the speed of processing and the number of pixels misclassified by the algorithm.

The paper is organized as follows. Section 2 presents a review of relevant differential geometry results, and Section 3 presents a review of MRF and the Gibbs Distribution (GD). Our algorithm will be given in Section 4. Section 5 shows results of processing various range images and Section 6 outlines the conclusions.

2. H AND K CURVATURE PARAMETERS

H and K are identified as the local second order surface characteristics that possess several invariance properties and represent extrinsic and intrinsic surface geometry features respectively. The sign of these surface curvatures can be used to classify the image surface points into one of eight basic types. Fig.1 shows the corresponding surfaces labels. These two curvature parameters can be calculated using [1] :

$$K=\frac{f_{xx}f_{yy}-f_{xy}^2}{(1+f_x^2+f_y^2)^2}$$

$$H=\frac{f_{xx}+f_{yy}+f_{xx}f_y^2+f_{yy}f_x^2-2f_xf_yf_{xy}}{(1+f_x^2+f_y^2)^{3/2}}$$

	K>0	K=0	K<0
H<0	Peak	Ridge	Saddle Ridge
H=0	---	Flat	Minimal surface
H>0	Pit	Valley	Saddle Valley

Fig.1 Surface type labels from H and K.

Some of the problems with the HK-sign map are :a) Preliminary smoothing is necessary to obtain reasonable values for H and K [1]. However, after filtering the HK-sign surface labels then reflects the geometry of the smoothed surface data. Hence, the HK-sign map must be further processed, b) In the presence of noise HK-sign map surface labels tend to connect the labels of neighborhood, but distinct, surface regions. c) Global surface properties is lacking.

3. MRF AND THE GD

The concept of a MRF is a direct extention of the concept of a Markov process to higher dimension [8]. A discrete MRF on a finite lattice is defined as a collection of random variables, which correspond to the sites of the lattice.

Definition of MRF:

Consider an N X N rectangular lattice, and r= (i,j) be an index of pixel locations, where i,j specify pixel row and column location and satisfy $1 \le i,j \le N$. Let $\{x_r\}$ denote a random field, with x_r the field at pixel r, X a vector specifying the field over an entire N X N lattice and having components x_r, and $X_{(r)}$ the field everywhere but at pixel r. Then $\{x_r\}$ is a MRF if

$$p(x_r|X_{(r)})=p(x_r|x_v,\ v \in D_p)$$

for all r, and P(X=x)>0 for all x. D_p denotes a neighbor set,

$D_p=\{v=(l,m)$ such that $||r-v||^2 \le Np$ and $v \neq r\}$

Where P is the order of the process, and N_p is an increasing function of P. N_p takes the values 1,2,4,5,8 for P=1,2,3,4,5, respectively (N_p is the square of the euclidean distance to the farthest neighbor).

Definition of cliques:
Given a set of neighborhood on a lattice a clique c is such that;
a) c consists of a single pixel, or
b) for $r \neq v$, $r \in c$ and $v \in c$ implies that r and v are neighbors. The collection of all cliques is denoted by C. The neighborhood system up to the fifth order and the cliques associated with the first order system are shown in Fig.2.

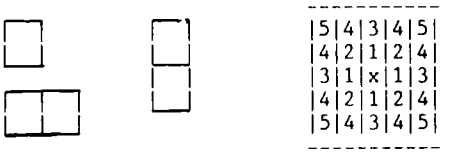


Fig. 2 The neighborhood systems and the cliques for the first order.

Definition of GD:
A random field $X=\{x_r\}$ defined on a lattice has an associated GD (or equivalently is a Gibbs Random Field(GRF) with respect to D_P iff its joint distribution is of the form:

$$P(X=x) = (1/Z)*\exp(-U(x))$$

where $U(x)=\sum_{c\in C} V_c(x)$ is the energy function,

V_c=potential associated with clique c,
and $Z=\sum_x \exp(-U(x))$ is a normalization factor.

Hammersley-Clifford theorem:-Let D_P be a neighborhood system on a finite lattice . A random field X is a MRF with respect to D_P iff its joint distribution is a GD with cliques associated with D_P.

4. OUR ALGORITHM

Biological vision systems achieve efficient, robust and reliable recognition in highly variable environments through the integration of many visual sources. For example the simple task of locating objects boundaries can be performed far more effectively by integrating evidence of discontinuities in image intensity, stereo disparity, speed and direction of motion and texture information than by using evidence from a single visual source on its own. The integration problem is computationally complex. The integration can be achieved by associating a MRF on a lattice to each physical process and another (binary) model to its discontinuties. The lattice are coupled to each other to reflect the interdependence of the corresponding process in image formation. Similar work using this approach can be found in [6,9] among others. In general,the latter methods, are computationally expensive and the number of quantization levels must be small (typically 2 or 3). The use of H and K allows us to reduce the number of levels from 256 (the original image) to 3 levels (-,0,+ for H and K).

The flow chart of our algorithm is shown in Fig.3. The H and K are calculated in multi-scale fashion. Then the output of the multi-scale is combined with the edge information and the surface normals. This will give us a seed region and edge information which will be entered to the region growing algorithm. H and K are processed separately and then combined to obtain the HK-sign map. Final surface description of the object can be obtained by fitting surfaces to the HK-sign map.

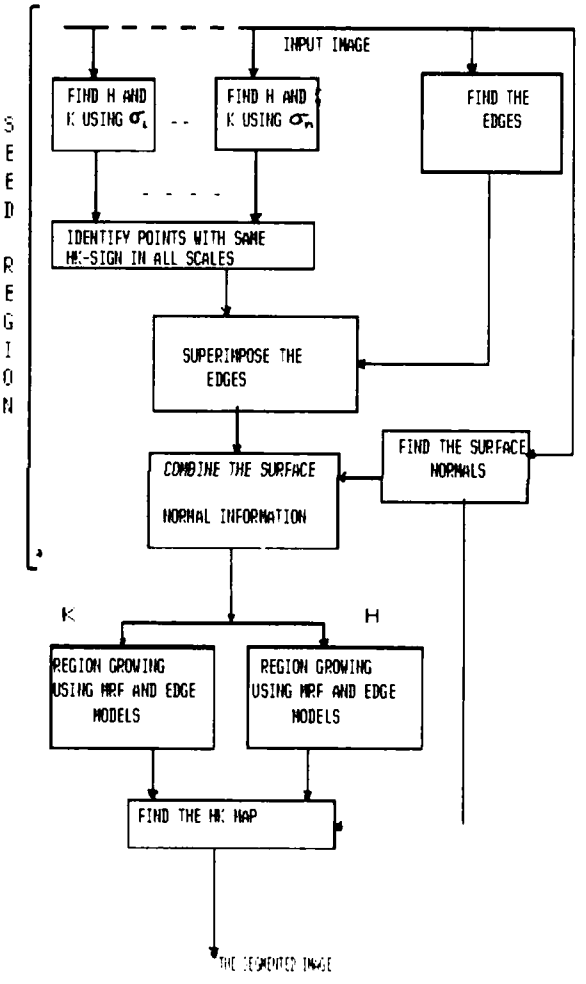


Fig 3 The algorithm flow chart

4.1 Finding the Seed Region:

The seed regions are obtained using a multi-scale approach. This approach is justified because the output from different scales is going to change significantly on the boundary of the object while the points well inside the surface will not change. The input image is smoothed with a Gaussian filter of different standard deviation and

for each output the values of H and K are estimated. The sign of the resulting H and K values are then used to form a three level image for H and K. The outputs from the multi-scale are then combined by identifying the points on the different scales (3 in our experminits) where the value of H and K signs are identical. The labeling of these points are assumed to be correct and is used as the seed region for the region growing algorithm. The edges are also obtained and superimposed on the the multi-scale output.

In some cases, for example for a roof edge, surface normal information is needed to segment the planar regions. The surface normal is estimated by fitting a plane in a 3X3 mask size. The surface normal information is also used to segment the background of the object.

4.2 Region Growing:

The seed region output is entered to the region growing step. The region points are modeled as a MRF with variable neighborhood size. For the edge points, an additional term is used. The energy function integrates these two models :

$$U(f,e;g)=\sum_{D_P} V(e)+(1-e)*\sum_{D_P} V(f_r,f_v|e)$$

where g is the output from the seed region step, e is the edge point (binary), f is the processed image, D_P is the neighborhood system, V(e) is the energy function due to the presence of edge. V(e) is computed by using Fig.4 in which a value is assigned for V(e) based on all the possible local configurations of the edge point. This model encourages the formation of continuous edges and discourages thick edges. For example if points B and C are edge points the model discourage the presence of edge at A.

$V(f_r,f_v|e)$ is the energy function due to the pixel f_r label in the neighborhood area given the edge points. Only the single pixel clique is used in the experimental results.

A	B	C	D	V(e)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

* *
C D

* *
A B

0 noedge
1 edge

Fig.4. The Edge Model.

The region growing algorithm proceeds by collecting the edge points and the pixels unclassified by the multi-scale approach in an array. A point is then picked at random. The energy function given earlier is then minimized in the local area surrounding the selected pixel. This is repeated for all the points in the array for a number of iterations (maximum of 30 was used in the experimental results).

5. EXPERIMENTAL RESULTS

The algorithm has a good parallel computational structure, since the multi-scale, edge detection and the surface normal estimation can be computed simultaneously. Also the computation of H and K are independent and can be computed simultaneously. The algorithm has been tested an a number of synthetic and real images. The images are 128X128 with 8 bits/pixel. The H and K values are obtained following the procedure suggested be [1]. Experimental results for different objects are shown in Fig.5 through 7. To assess the importance of the edge information,images are processed with and without the edge model. We have used a variable neighborhood systems (up to the fifth order) for the region model.

The first object (Fig.5a) is a synthetic image of a sphere. The output of the seed region step is shown in Fig.5b for H and in Fig.5e for K. The result of the region growing step is shown in Fig.5c for H and in Fig.5d for K. Fig.5f shows the final HK-sign map obtained by combining the output from Fig.5c and Fig.5d. As can be seen the image is segmented perfectly using this method. Fig.6a is a range image of a coke bottle obtained using a laser range finder at the Enviromental Research Institute of Michigen (ERIM). The content of Fig.6 are similar to Fig.5. Good segmentation is obtained with the exception of a small area at the tip of the coke bottle.

Fig.7 shows the results for a coffee cup obtained from ERIM. Fig.7a shows the image. Fig.7d and Fig.7h show the seed region obtained for H and K respectively. The range image is then processed in two different ways. Fig.7e and Fig.7i show the output of the region growing algorithm with the edge model. Fig.7b shows the final HK-sign map. The segmentation results obtained were good with the exception of the handle of the coffee cup, which was not classified. This is because of the size of this region and the restriction in the algorithm on the number of pixels required for classification. In Fig.7f and Fig.7g the outputs of the region growing algorithm without the edge model are shown. Fig.7c shows the final HK-sign-map. In this case the handle is classified as planar region, also small regions of the cylindrical surfaces of the object are classified as planar. A comparative study of Fig.7b and Fig.7c illustrates that inclusion of the edge model leads to less misclassified points.To emphasize the advantage of using a variable neighborhood system for the MRF. Fig.8 shows the results of processing the coffee cup with different fixed neighborhood systems. In this figure the time required for processing is compared for five different neighborhood systems. The time required for the variable neighborhood system (up to the fifth order) is also shown.

The other graph in the figure shows the difference in the classification between the variable and the fixed neighborhood system MRF for a fixed number of iterations (30). Thus the use of the variable neighborhood system gives a good compromise betwe the time required for processing and the number of misclassified pixels.

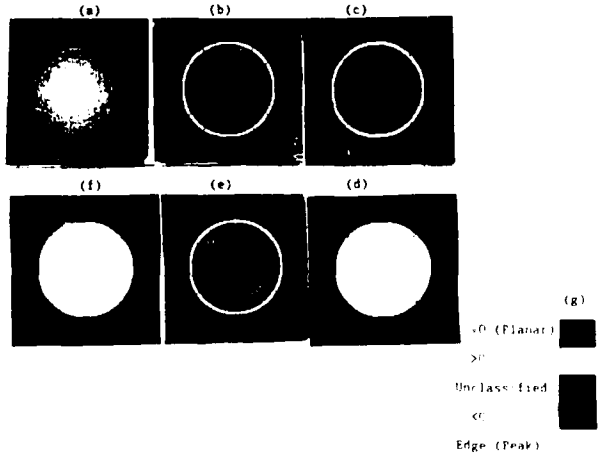


Fig. Results of Processing a Synthetic Image - A sphere.

a)The original range image.
b)H seed region multi scale output (black:Unclassified).
c)H region growing output.
e)K seed region multi scale output (Black: unclassified)
d)K region growing output.
f)HK sign map.
g)Colour codes for Fig.5c and 5d.
Edges are superimposed on Fig.5b through 5f.

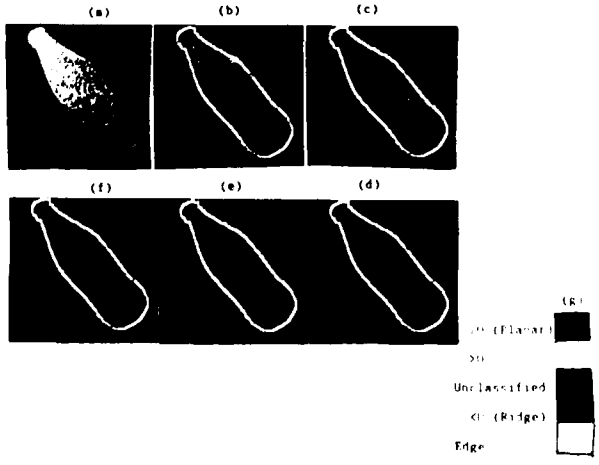


Fig 6 Results of Processing a Synthetic Image - A Coke Bottle

a)The original range image.
b)H seed region multi scale output (Black Unclassified)
c)H region growing output.
d)K seed region multi scale output (Black unclassified)
e)K region growing output
f)HK sign map
g)Colour codes for Fig 6c and 6e
Edges are superimposed on Fig 6b through 6f

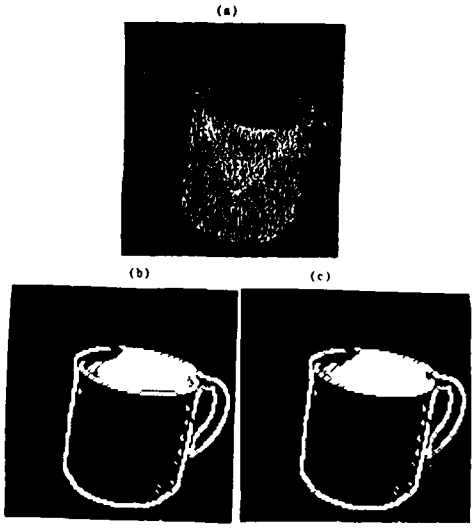


Fig. 7 Results of processing a range image - A Coffee cup.

a)The original image.
b)The HK-map with edge model.
c)Similar to (b) without edge model.
d)H seed region multi-scale output (Black:Unclassified).
e)H region growing output with edge model.
f)Similar to (e) without edge model.
h)K seed region multi-scale output (Black:Unclassified).
i)K region growing output with edge model.
g)Similar to (i) without edge model.
k)Colour codes for Fig.7e,7i,7i and 7g (left) and for Fig.7b and 7c (right).
Edges are superimposed on Fig.7b through 7g.

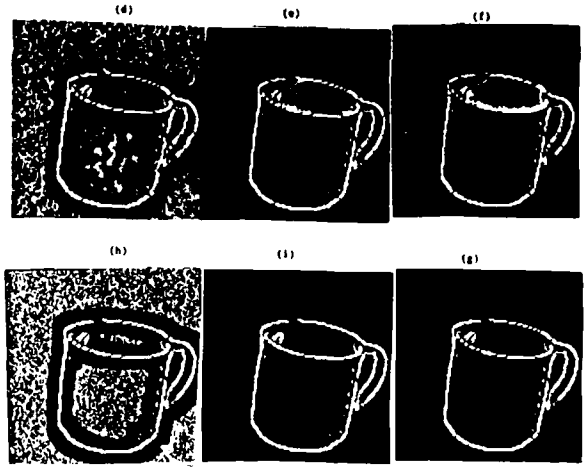


Fig (7)Continued

6. CONCLUSION

An algorithm for segmentating range images using a variable neighborhood system MRF and edge models is presented. This approach allows us to integrate a number of surface characteristics in an efficient way. The results shown are good with the exception of the handle of the coffee cup. The use of H

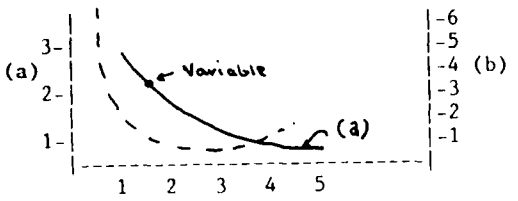


Fig.8 Comparsion between fixed and variable neighbohood system a)Time (Min), b)Difference in classification (1000 Pixels).

and K allow us to work with a small number of levels (3 compared with 256) which makes the processing faster. The use of variable neighborhood system MRF reduces the number of misclassified pixels with a small increase in the time required for processing.

The future work will concentrate on a surface fitting step which will be used to obtain a final description of the range data. This will also help in classifying the unclassified pixels in the output of the algorithm. Also the algorithm relies on the initial seed region calculation using the multi-scale approach. A possible variation to the procedure and to the parameter estimation will be studied.

REFERENCES

[1] P.J. Besl, "Surfaces in Early Range Image Understanding," Ph.D. Dissertation, Dep. Elec. Eng. Comput. Sci., Univ. of Michigan, Ann Arbor, Rep. RSD-TR-10-86, Mar. 1986.

[2] F. Cohen, and D.B. Cooper, "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields," IEEE Trans. Pattern Anal. Machine Intell., vol.9, no.2, pp. 195-219, Mar. 1987.

[3] P. Chou,"Multi-Model Segmentation Using Markov Random Fields," in Proc. Int. Joint Conf. Artificial Intell., July 1987.

[4] G.R. Cross, and A.K. Jain,"Markov random field texture models," IEEE Trans. Pattern Anal. Machine Intell., vol.5, no.1, pp. 25-39, Jan. 1983.

[5] H. Derin, and H. Elliott,"Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields," IEEE Trans. Pattern Anal. Machine Intell., vol.9, no.1, pp. 39-55, Jan. 1987.

[6] S. Geman, and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and Bayesion Restoration of Images," IEEE Trans. Pattern Anal. Machine Intell., vol.6, no.6, pp. 721-741, Nov. 1984.

[7] F.R. Hansen, and H. Elliott,"Image Segmentation Using Simple Markov Field Models," Comput. Graphics Image Processing, vol.20, pp. 101-132, 1982.

[8] R. Kindermann and J.L. Snell, "Markov Random Fields and their Applications," vol.1, Amer. Math. Soc.(1980).

[9] J.L. Marroquin, "Probabilistic Solution of Inverse Problems," M.I.T. AI Lab., Cambridge, MA, Tec. Rep. 860, 1985.

Compression of Image Data Using Arithmetic Coding

by

Ahmed Desoky, Carol O'Connor and Thomas Klein
Department of Engineering Mathematics and Computer Science
University of Louisville
Louisville, Kentucky 40292

ABSTRACT

The purpose of this paper is to measure the amount of compression that can be accomplished by the use of Arithmetic Coding and the coding processing time. An IBM-PC based system has been developed for both encoding and decoding. The results using adaptive and non-adaptive techniques are presented. The test data consisted of a 256 gray level image file and seven classes of different data files. Performance evaluation is discussed in terms of encoding time and decoding time.

I. Introduction

Minimum redundancy codes of a data system is attractive for two major reasons: storage saving and performance improvement. Storage saving is a direct and obvious benefit, whereas performance improvement is the direct result from the fact that less data are moved in the case of communication. Arithmetic coding and Huffman coding are approximately minimum redundancy coding techniques where code words are of variable-length.

Huffman coding is one of the pioneering works in the construction of minimum redundancy code. It was developed in 1952 by Huffman [1]. Because of its simplicity, it has been developed on small systems with encouraging results [2]. To code a file using the standard Huffman method:

1. Determine the frequency of each character.
2. Construct Huffman coding table by assigning variable length-codes to each character. Generally, this results in the assignment of short codes to characters that occur most frequently.
3. Encode the input file.
4. At any future time, the file can be reconstructed using the stored Huffman coding table.

Arithmetic coding [3] has been proposed as being more superior in most respects than the Huffman scheme. Here, the input message is represented as an interval of real numbers between 0 and 1. The longer the message, the smaller the interval needed to represent it, and thus more bits are needed to describe the interval. An individual symbol of the message reduces the size of the interval by an amount determined by its frequency of occurrence. The more likely symbol reduces the range by less than an unlikely one, and consequently adds fewer bits to the coded message. The end of the message is represented by a unique message terminator symbol.

Arithmetic coding technique was introduced in a textbook by Abramson [4]. As a compression technique, this method is not widely known. However, reference [5] is a good introduction to the subject of arithmetic coding.

II. Algorithms and Implementation

Both the encoder and the decoder know (or can generate) the probabilities of occurrences of, and the portions of the range occupied by, the various symbols, and the initial range is [0,1). With this in mind, the decoder can deduce the encoded characters one by one by analyzing which range the interval lies within as each symbol is revealed. Also, both encoder and decoder know a unique_eof_symbol that will be used to terminate messages.

The encoding and decoding algorithms can be summarized as follows:

```
ENCODE
while not eof
begin
  read symbol
  if eof
     symbol = unique_eof_symbol;
  current_range = range_high - range_low;
  range_high = range_low + current_range *
  frequency_sum[symbol - 1];
  range_low = range_low + current_range *
  frequency_sum[symbol];
end;
```

```
DECODE
while symbol <> unique_eof_symbol
begin
  read code_value;
  symbol = 1;
```

```
  while not (frequency_sum[symbol] <=
  (code_value - range_low) / (range_high -
  range_low) < (frequency_sum[symbol - 1])
     symbol = symbol + 1;
  if symbol = unique_eof_symbol
     stop;
  current_range = range_high - range_low;
  range_high = range_low + current_range *
  frequency_sum[symbol - 1];
  range_low = range_low + current_range *
  frequency_sum[symbol];
  write symbol;
end;
```

In the above pseudocode, the possible symbols are numbered 1, 2, ..., number_of_symbols + 1, with the last symbol being the unique_eof_symbol. The frequency range of an individual symbol is:

```
[frequency_sum[symbol] ,
                frequency_sum[symbol - 1])
```

In practice, we first generate a list containing probabilities for each symbol that can be encoded. Since each symbol is a byte, the set pf symbols can be represented by integers from 0 to 255, with the unique_eof_symbol being 256. We represent this scheme as a 257 position array. The frequencies can be represented in one of two ways; a non-adaptive or fixed representation in which the frequencies are determined in advance for a given class of data files to be compressed (i.e. image files), and an adaptive method in which the frequencies are generated based on the symbols observed during compression or expansion of the file being processed. We will see that the adaptive model in general provides more desirable results than the fixed model. Probabilities are represented as integers and cumulative counts are stored in a second array. To prevent overflow the counts are scaled as necessary.

Since all values are represented as integers and operations are performed on only a byte at a time, all data must be transmitted and received incrementally. To accomplish this, bits in the low and high ends of the range are transmitted as they become the same and the range is rescaled.

To begin the encoding process a byte is read and is used as an index to the frequency array. In the non-adaptive or fixed case the frequency is simply read from the array. In the adaptive case the current frequency is used and the the frequency is updated to include the new symbol. Next the frequency is applied to the range and the next byte is read. A code buffer is maintained to hold bits to be transmitted. When the byte_long buffer is filled, the byte is written then cleared to accept new data. When end_of_file is reached, the unique_eof_symbol indicator is encoded and the encoding process is complete.

To decode, a byte of the encoded file is read and the first code value is extracted. the frequency array is then scanned to find the symbol corresponding to that code. In the adaptive version the frequency list is then updated to include the new symbol. The symbol is written and the remaining portion of the byte is processed to extract the next code value. When the byte has been exhausted, the next byte is read and the process continues until the unique_eof_symbol is identified. At this point any extraneous bits are written and the decoding process is complete.

FIGURE 1

TEST DATA BYTE VALUE FREQUENCIES

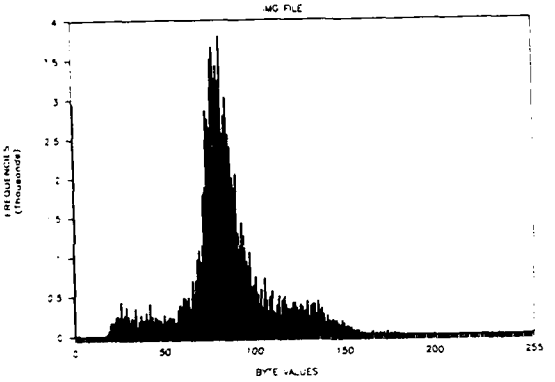


FIGURE 2

TEST DATA BYTE VALUE FREQUENCIES

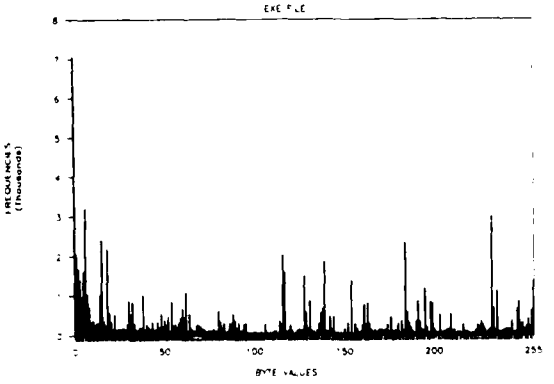


FIGURE 3
TEST DATA BYTE VALUE FREQUENCIES

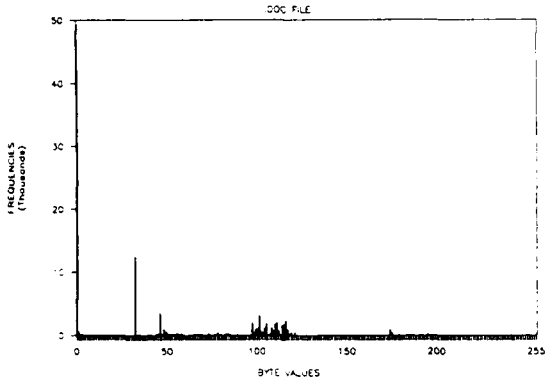
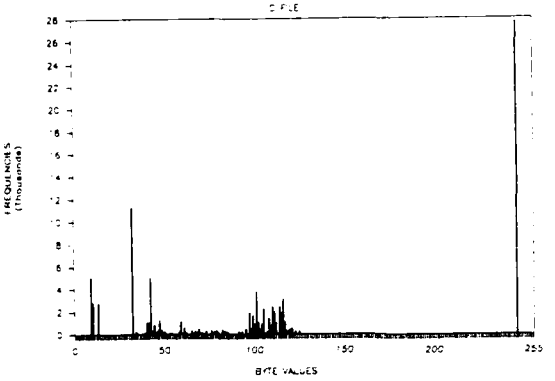


FIGURE 4
TEST DATA BYTE VALUE FREQUENCIES



III. Results and Discussion

The results of the experiment are base on the application of both fixed and adaptive Arithmetic Coding to eight classes of 100,000_byte files. The files were an executable binary file, a C source code file, a Multimate document file, a 256 gray level image file, an ascii data file, a dBase III data base file, a text file and a Lotus 123 spread sheat file. The frequency distributions of four of these file are presented in Figures 1 through 4.

The program implementation is in Microsoft Quick C on an IBM PC XT 286 running DOS 3.2. Times are in seconds per byte of uncompressed data and include all I/O and operating system overhead.

Three different distributions were used for the non_adaptive or fixed method. The first, a frequency distribution similar to that of the English language, the second, a distribution generated by averaging the actual frequencies of the eight test files, and the third a distribution generated from the image test file (Figure 1). In the adaptive method the frequency distribution is dynamic, changing as each symbol is observed. The results of encoding and decoding of each of the eight data files are demonstrated in Tables I through IV.

Not surprisingly, the results in Tables I through IV reflect that the English language distribution performed best for the files containing English_like data and the image distribution performed best for the image file. The average distribution performed surprisingly well. However, the adaptive method consistently performed better than the other distributions.

IV. References

[1] Huffman, D. A.: A method for the construction of minimum redundancy codes. Proc. Inst. of Elect. Radio Eng. 40, 9 (Sept. 1952), 1098 - 1101.

[2] Desoky, A., Gregory, M.: Compression of text and binary files using adaptive Huffman coding techniques. Proc. IEEE Sotheastcon '88 (April 10 - 13, 1988), 660 - 663.

[3] Rubin, F.: Arithmetic stream coding using fixed precision registers. IEEE Trans. Inf. Theory IT-25, 6 (Nov. 1979), 672 - 675.

[4] Abramson, N.: Information theory and coding. McGraw-Hill, New York, 1963, (pp. 61 - 62).

[5] Langdon, G.G.: An introduction to arithmetic coding. IBM J. Res. Dev. 28, 2 (Mar. 1984), 135 - 149.

TABLE I

Fixed Model English Language Distribution

	Output	Encode time		Decode time	
	(bytes)	Total	per byte	Total	per byte
.EXE file	149,793	211.0	0.002110	324.0	0.003240
C source program	86,052	135.0	0.001350	202.0	0.002020
Multimate .DOC file	113,341	193.0	0.001930	208.0	0.002080
EMCS619 .IMG file	126,755	190.0	0.001900	259.0	0.002590
ASCII data file	111,755	187.0	0.001870	226.0	0.002260
dBase III .DBF file	109,781	184.0	0.001840	225.0	0.002250
Text file	68,737	130.0	0.001300	189.0	0.001890
Lotus 123 .WKS file	143,727	210.0	0.002100	246.0	0.002460

TABLE II

Fixed Model Average Distribution

	Output	Encode time		Decode time	
	(bytes)	Total	per byte	Total	per byte
.EXE file	106,376	162.0	0.001620	262.0	0.002620
C source program	78,056	128.0	0.001280	194.0	0.001940
Multimate .DOC file	56,258	112.0	0.001120	146.0	0.001460
EMCS619 .IMG file	98,062	172.0	0.001720	237.0	0.002370
ASCII data file	68,846	131.0	0.001310	168.0	0.001680
dBase III .DBF file	67,555	129.0	0.001290	167.0	0.001670
Text file	72,969	138.0	0.001380	199.0	0.001990
Lotus 123 .WKS file	69,450	136.0	0.001360	162.0	0.001620

TABLE III

Fixed Model Image Distribution

	Output	Encode time		Decode time	
	(bytes)	Total	per byte	Total	per byte
.EXE file	131,041	206.0	0.002060	288.0	0.002880
C source program	108,047	164.0	0.001640	234.0	0.002340
Multimate .DOC file	84,206	156.0	0.001560	175.0	0.001750
EMCS619 .IMG file	79,534	145.0	0.001490	216.0	0.002160
ASCII data file	109,571	183.0	0.001830	213.0	0.002130
dBase III .DBF file	107,606	169.0	0.001690	212.0	0.002120
Text file	106,758	164.0	0.001640	236.0	0.002360
Lotus 123 .WKS file	101,103	163.0	0.001630	195.0	0.001950

TABLE IV

Adaptive Model

	Output	Encode time		Decode time	
	(bytes)	Total	per byte	Total	per byte
.EXE file	87,382	152.0	0.001520	183.0	0.001830
C source program	61,510	121.0	0.001210	134.0	0.001340
Multimate .DOC file	42,371	91.0	0.000910	90.0	0.000900
EMCS619 .IMG file	79,522	139.0	0.001390	151.0	0.001510
ASCII data file	52,696	103.0	0.001030	99.0	0.000990
dBase III .DBF file	50,976	100.0	0.001000	97.0	0.000970
Text file	56,209	100.0	0.001000	107.0	0.001070
Lotus 123 .WKS file	50,909	94.0	0.000940	104.0	0.001040

IMAGE ANALYSIS OF THE MICROVASCULAR SYSTEM IN THE RAT CREMASTER MUSCLE

Carol O'Connor, University of Louisville
Ahmed Desoky, University of Louisville
Cathy Senft, University of Louisville
Patrick Harris, University of Louisville

ABSTRACT

A VAX-based image processing system has been developed for the digitization and analysis of the microvascular system in the rat cremaster muscle. These are images of blood vessels which are less than one millimeter in diameter. The purpose of this system is to obtain quantitative morphometric data on the microvascular system which cannot be easily obtained by manual methods. Animal studies have shown that microcirculation can be used in the detection of certain systemic vascular diseases such as diabetes mellitus and hypertension. These diseases involve major disturbances in the dimensions and the distributions of microvessels. A similar phenomenon occurs with the introduction of substances such as hormones into the system. The developed techniques will be used to determine the blood vessel distributions for a number of samples. Statistical testing will then be done on samples of images comprising diseased and nondiseased animals, and on samples of before and after introduction of compounds, to determine which image component parameters best discriminate diseased and nondiseased samples, and best describe the effects of the compounds on the microvascular system.

1. INTRODUCTION

The Center of Applied Microcirculatory Research has recently been established at the University of Louisville, with Dr. P. D. Harris as its director. The primary purpose of the Center is to develop microcirculation medicine as a new applied discipline. Microcirculation Medicine is a new clinical arena with focus on the structure, function, pathology, and therapy of blood vessels less than one millimeter in diameter. Several relevant factors entered into the creation of the Center at the University.

Scientific literature documents that microvessels, at different levels in the same organ, function in different ways for various purposes [1,4,10,16]. Microvascular levels respond differently to hormones, disease processes, and therapeutic agents and procedures [1,5,6,11,12, 13,14,16]. There has been a tremendous increase in knowledge, techniques, and understanding of microcirculatory mechanisms resulting from animal studies during the past 20 years, and studies on animal models of human diseases have amply shown that microvascular events play an important role in the development of some of these disease processes [10].

Secondly, clinical sciences now use little of this expanding microcirculatory knowledge [8,9]. The few approaches investigated, such as observing the human microcirculation in specialized tissues such as the conjunctiva of the eye and the nailfold of the fingers or toes [2,7,9,15], have not been able to provide useful data on microvascular function in humans. These approaches have not affected the outcome of clinical medicine, with one dramatic exception, which is described to demonstrate the importance of this research for clinical medicine.

In the mid 1960's there was a severe epidemic of infectious meningitis in China, with a 90% mortality rate for children under 2 years of age. The treatment was a Chinese herbal drug labeled "654," whose toxic level is only slightly higher than its effective treatment level. Thus, many children treated subsequently died from "654" toxicity. In 1965, a young Peking clinician, Dr. Rui-juan Xiu, put together a simple bedside microscope to assess the "quality" of blood perfusion in the nailfold capillaries of children. She used this device to adjust the infusion rates of "654" to maintain an effective but non-toxic therapeutic dose in each sick child. This individualized control of "654" therapy reduced the infant mortality rate in infectious meningitis to less than 10% within a three month period.

The Center emphasizes multidisciplinary research teams, including researchers from clinical medicine, basic health sciences, and engineering. Collaborations had been developing between many researchers in these fields, and this strong nucleus aided in the development of the Center at the University of Louisville.

The Center has established six project areas initially in which to concentrate its efforts. Several of these involve image processing and pattern recognition, and are described briefly. Under certain conditions, the venules exhibit a tendency to leak; hormones can cause holes in the venules, or abnormalities of the small veins in tumors can contribute to this leakage phenomena. Using image analysis of images of the microvascular system under these conditions, the goals are to measure the amount of leakage, the average rate of leakage, where the leakage is occurring, and the effect of dosage on the leakage.

In the microvessels, white cells are in free flow. However, white cells may stick to the vessel walls, roll along the vessel walls, and may clump to one another. This stickiness is an early sign of leukocyte activation during conditions such as infection, tissue transplant rejection, and systemic vascular diseases. Image analysis will be used to study this stickiness phenomenon in terms of measuring the tendency of the white cells to stick, their clumping tendency, how rapid the clumps grow, the number of clumps that pass a certain point over time, and, if they stick to the wall, for how long. The general methods developed here can also be used to study emboli and thrombi.

Using nailfold images of the capillary system, image analysis will be used to measure

the velocity in the nailfold loops, based on plasma gaps, determine the nailfold microcirculatory characteristics in various disease categories, and identify image-analysis sequences for quantification of desired microvascular parameters in nailfold microcirculation.

2. CURRENT PROJECT

The arterioles in the microvascular system are very dynamic in terms of dilating or constricting. This has many causes: hormones, therapeutic agents, and many systemic vascular diseases such as hypertension and diabetes mellitus. It is this "diameter phenomena" which is the concern of this project. In general, the goals of this project are to use image analysis techniques to measure the diameters and lengths of the arterioles in a certain region in a tissue, and determine sequential microvascular changes in these parameters from various causes. Systemic vascular diseases involve major disturbances in microvessels which range from 1 millimeter (the large arterioles) down to 0.003 mm (the smallest arterioles). Animal studies have suggested that detectable microvascular pathology appears very early in the development of some forms of diabetes mellitus [2], have shown pathology in the artery wall at an early stage in the development of several experimental forms of hypertension [10], and that treatment may reverse these microvascular disturbances at least in the early stages. Thus, animal studies suggest that observations of microvascular changes in humans can provide very early detection of some forms of systemic vascular disease, as well as assessments of the efficacy of early therapeutic interventions.

Presently, microcirculation in the nailfold has been studied, and provides useful information on capillary perfusion [7]. However, this system does not visualize the arterioles and venules which are involved early in systemic vascular disease. Studies have also examined the microcirculation in the conjunctiva (white of the eye). However, this has provided data only for a sparsely arranged vessel network, with little parallelism between large arterioles and venules; whereas the general systemic microcirculation contains parallel and adjacent large arterioles and venules. Thus, a new method for microvascular observation that is truly representative of the general microcirculation is needed in humans to detect and to assess the treatment of systemic vascular disease.

Currently, this project involves image analysis of the microvascular system for a region of the rat cremaster muscle. This thin muscle tissue allows transmitted light microscopy and epi-illumination fluorescent microscopy simultaneously. A closed-circuit television system, as shown in Figure 1, can be used to obtain video images of the microvascular system. A new gingival preparation [3] for observation of the microcirculation in the gum and lip area of the mouth is also being developed for use in humans. The gingival microcirculation has arterioles, capillaries, and venules which can also be observed by microscopy, and this circulation is typical of many other body tissues. Thus, it is envisioned that when the image analysis research is completed on the rat cremaster muscle, it can be extended to humans using the gingival preparation.

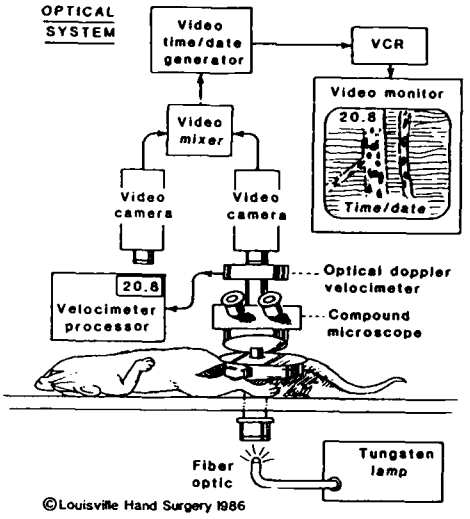


Figure 1: Optical System for obtaining images of microvascular system in the rat cremaster muscle.

3. METHODOLOGY

Videotapes are supplied by the basic health scientists working with the Center showing the microvascular system of the rat cremaster muscle for normal rats, rats bred for systemic diseases such as hypertension, and for the microvascular system before and after application of a substance such as a hormone. Images from the videotapes are obtained using the VAX Vision System. The hardware component of the VAX Vision System is the ITI-IP-512 digital image processing system, whose fundamental components are a video digitizer and a frame buffer. The frame buffer contains 256K bytes of high-speed random access memory, where 512 x 480 pixel image frames are stored. Individual pixels are 8 bits deep, allowing for 256 gray levels. With this system, a standard RS-170 video signal can be digitized, stored, and displayed on a video monitor in real time. The software of the VAX Vision System consists of VISION-SUBS, a series of subroutines for controlling the ITI vision hardware; VAXTIPS, an interactive image manipulation system; and VISCOM, a stand-alone program which must be run whenever the VMS operating system is booted.

Using the VAX Vision System, a frame from the videotape is grabbed, digitized, and stored for further processing. A typical image is shown in Figure 2. The parameters of interest for this project are the arteriole lengths and diameters in this region of the rat cremaster muscle. To obtain the measurements of interest, standard thresholding techniques were first applied to better define the vessels. However, problems arose with this approach. The image contrast is not very good, and more importantly, no automatic procedure could be developed to

distinguish the arterioles (which are the vessels being studied) from the venules (which are not wanted in this project). Currently, the approach is to use manual cursor movement to mark points along the edges of the arterioles of interest. These points are then connected, creating an outline of the arterioles. A binary image with the arterioles filled in is then produced, to which a thinning algorithm is used to obtain a skeleton of each arteriole. Starting with an original image as in Figure 2, Figures 3 through 5 demonstrate the output of each of these stages.

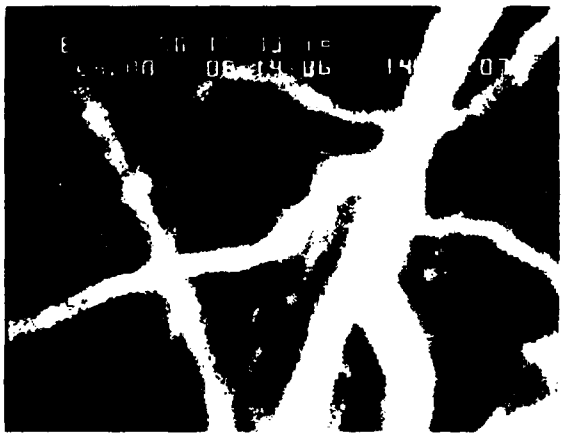


Figure 2: Original image of microvascular system in a region of the rat cremaster muscle.

Figure 3: Image after marking and connecting of arteriole wall.

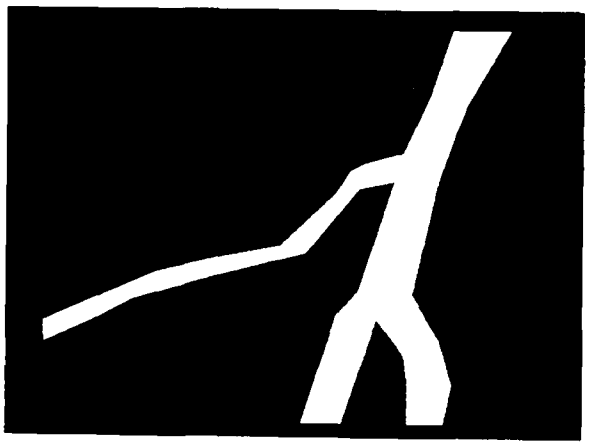


Figure 4: Binary image of filled-in vessel

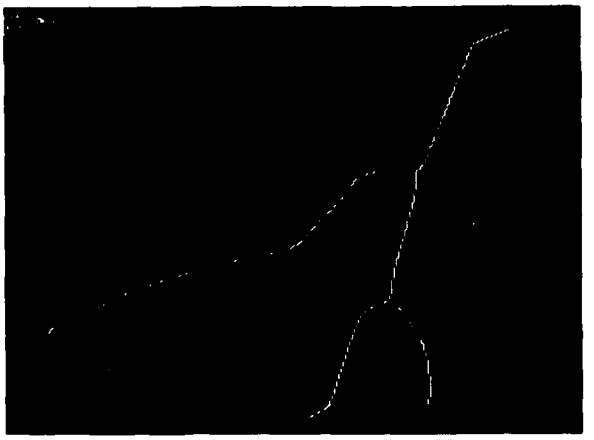


Figure 5: Skeleton image of vessel after thinning

In pseudocode, the filling and thinning algorithms can be summarized as follows:

```
FILL (NUMBER OF ROWS, NUMBER OF COLUMNS, OBJECT
      COLOR)

begin
     repeat for each pixel in IMAGE, row by row
     and column by column
       if IMAGE(ROW, COLUMN) = white
         if   surrounding   pixels   fit   corner
         pattern (See Fig. 6)
           SEED ROW  - row of pixel inside corner
           SEED COLUMN  - column of pixel inside
                          corner
           exit repeat loop
         else
           skip to beginning of next row
         end if
       end if
     end repeat


     call CONNECT (SEED ROW, SEED COLUMN, LABEL
                   TAG,  NUMBER  OF  ROWS,  NUMBER
                   OF COLUMNS)
     set every LABELed pixel to white

end


CONNECT (ROW, COLUMN, LABEL TAG, NUMBER OF ROWS,
         NUMBER OF COLUMNS)

begin
     if ROW and COLUMN are within image limits
       if IMAGE(ROW, COLUMN) = black
         set corresponding pixel on screen to
         white
         IMAGE(ROW, COLUMN)   - LABEL TAG
         call CONNECT with the coordinates of
         the four neighbor (See Fig 7) pixel
         coordinates (the other parameters stay
         the same)
       end if
     end if
end
```

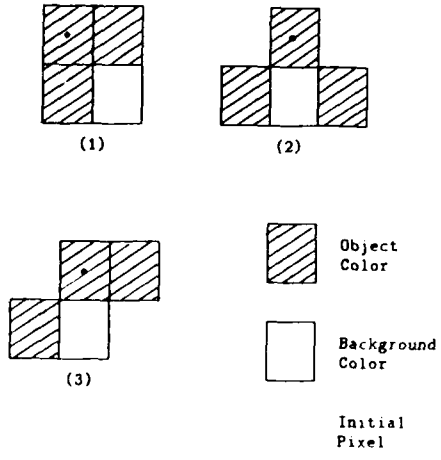


Figure 6: Corner patterns in fill algorithm

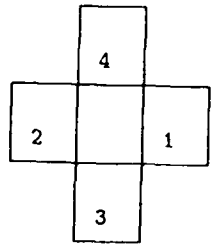


Figure 7: Four neighbors used in CONNECT

```
THIN2OBJ (IMAGE, NUMBER OF ROWS, NUMBER OF
          COLUMNS)
begin
     display image
     do while (edge pixels continue to be
               deleted)

       repeat for each pixel in IMAGE, row by
       row and column by column
         if IMAGE(ROW, COLUMN)   0
           DIRECTION  - 0  (See Fig. 8)

           do while (initial coordinates are not
                     reached again)

             if there are 2 to 6 white neighbors
             in the 8 neighbors
               if there is exactly one black to
               white transition in the neighbors
                 if pixel is a right or bottom
                 associated pixel
                   mark IMAGE(ROW, COLUMN) to be
                   deleted
                 end if
               end if
             end if

             ROW  - next edge pixel row
                    coordinate
             EDGE COLUMN  - next edge pixel
                            column coordinate
           end do while

           exit repeat loop

         end if
       end repeat
       repeat for each pixel in IMAGE
         if IMAGE(ROW, COLUMN) is marked to be
         deleted
           IMAGE(ROW, COLUMN)  - 0 (black)
         end if

       repeat both above repeats for left and
       top associated pixels

       display image
     end while

end
```

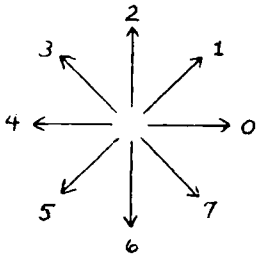


Figure 8: Direction labels in thinning algorithm

At the skeleton stage, the segments of the arterioles need to be defined. A measuring procedure will then find the length and width for each segment (or branch) of the arterioles of the original image. A segment begins at an end point and ends at another end point or at a branching point. An end point is a pixel with exactly one neighbor. The measure procedure searches the skeleton image from the upper left corner for the first end point. The skeleton is then followed until another end point or a branching point is reached. A pixel is considered a branching point if the skeleton forks or changes direction significantly. A fork is indicated by a white to black transisition for each branch at the fork in the surrounding neighbor pixels. A direction change is considered significant if the difference in horizontal or vertical coordinates from one pixel to the next changes in sign. For example, if the horizontal coordinate of pixels along the skeleton's path has been increasing and suddenly starts decreasing, the pixel where the change occurs is considered a branching point. This latter type of branching point check is needed because a fork branching point only exists when a single segment branches into two segments; if the skeleton has only two branches (a V shape), a fork branch does not exist. As a segment is followed, the pixels passed through are deleted from the skeleton image array. Any white neighbor pixels to a branching point pixel are also deleted. These deletions are necessary since the check for the next segment also starts from the upper left corner. The deletions at branching points separate the segments originating at forks, insuring an end point at the beginning of each branch. The previously detected branch will not be considered since it has been deleted.

Once a branch has been defined, measurements are made on it if it has a length of more than ten pixels. The length of a segment is given by the number of pixels in the skeleton for that segment. If the length of the branch is between eleven and twenty pixels, a width (diameter) measurement is taken at the midpoint of the branch. The midpoint is the pixel one half the lenth along the skeleton from the branch start point. All of the branches longer than 20 pixels have width measurements taken every 15 pixels, starting 15 pixels from the start point and ending 15 pixels from the end point. If there are not 15 pixels between the last measurement and the point 15 pixels from the end of the branch, a measurement is still taken at the latter point if there is a difference of at least 6 pixels along the skeleton from the last measurement.

The width measurement is defined as the minimum of the horizontal, vertical, positive and negative 45 degree, positive and negative 22.5 degree, and positive and negative 67.5 degree bisectors of the filled image through the coordinates of the measuring point. Each measurement is made by finding the length, in pixels, of a straight line between the first background points outside the filled image in opposite directions along the bisector from the skeleton point on the bisector. Each pixel along the bisector is tested for the background value. The positive and negative 45 degree points are found by moving one pixel horizontally and one pixel vertically until a black background pixel is reached. The positive and negative 22.5 degree points are approximated by moving in units of three pixels horizontally, one pixel vertically, two pixels horizontally, and then one pixel vertically. The resulting bisectors are actually at 21.8 and -21.8 degrees, but they are the closest approximations to the listed angles possible using a fairly small number of pixels. The positive and negative 67.5 degree bisectors are a similar approximation, whose units of three pixels vertically, one pixel horizontally, two pixels vertically, and one pixel horizontally produce bisectors actually at 68.2 and -68.2 degrees.
All width measurements for an image and the number of measurements with a particular width, regardless of location, are stored in a linked list. A graphical routine then uses this list to display a plot of the length versus width measurements.

4. DISCUSSION AND RESULTS

The algorithms described above to obtain length and diameter measurements of arterioles in the microvascular system of a region of the rat cremaster muscle have been implemented and are working. The graphical procedure has been implemented to obtain distributions of the total length of segments categorized by the diameter of the segment. While currently this entire procedure has been performed on a few select samples, the plan is to repeat this process for many image samples, including normal rats, certain diseased rats, and for before and after application of particular substances such as hormones. The changes in these distributions will then be analyzed. For example, one conjecture is that for a low dose of a hormone, the smaller diameter arterioles alone constrict, and the total length/diameter distribution will show a shift at the low diameter range only. As the dosage is increased, the diameters of arterioles affected increases. The long range goal is to correlate the shift/change in the total length/diameter distribution with an effective dosage level.

For hypertension, animal studies show pathology in the artery wall at an early stage in the development of several experimental forms of hypertension [10]. In hypertension, only large arterioles appear to be involved during the very early phase, and smaller arterioles are progressively involved at a later stage. Animal studies have also suggested that detectable microvascular pathology appears very early in the development of some, and maybe all, forms of diabetes mellitus [2]. The goal of this project is to use these changes in length/diameter distributions to monitor the progression of these diseases in humans, as well as to assess the efficacy of early therapeutic interventions.

In summary, this project uses image analysis techniques to obtain parameters (lenght, diameter) of the microvascular system in the rat cremaster muscle, with the goal being to correlate the changes in the length/diameter distributions (which classes of arterioles change size, and how much) with the dosage of a particular compound or the progression of some systemic vascular diseases. The techniques to obtain the needed parameter estimates have been developed, and soon various sample distributions will be obtained, analyzed, and compared. The gingival preparation will then be used to extend these techniques for humans.

REFERENCES

[1] Alsip, N., Harris, P.D., Asher, E.: Role of histamine in small arteriole dilation during hypoxia. Circ. Shock 21#4:343, 1897 (#134).

[2] Bollinger, A., Frey, J., Jager, K., Furrer, J., Seglias, J., Siegenthaler, W.: Patterns of diffusion through skin capillaries in patients with long term diabetes. N. Engl. J. Med. 305:1305-1310, 1982.

[3] Collins, J.G., Miller, F.N., Moore, R.L.: Non-invasive intravital flourescent microscopy of the hamster gingiva. J. Dent. Res. 66:315, 1987.

[4] Cryer, H.M., Garrison, R.N., Kaebnick, H.W., Harris, P.D., Flint, L.M.: Skeletal microcirculatory responses to hyperdynamic E. Coli sepsis in unanesthetized rats. Arch. Surgery 122:86-92, 1987.

[5] Cryer, H.M., Kaebnick, H.W., Harris, P.D., Flint, L.M.: Effects of tissue acidosis on skeletal muscle microcirculatory responses to hemorrhagic shock in unanesthetized rats. J. Surgical Res. 39:59-67, 1985.

[6] Cryer, H.M., Unger, L.S., Garrison, R.N., Harris, P.D.: Prostaglandin inhibition impairs renal microvascular blood flow responses during hyperdynamic bacteremia. Circ. Shock 21#4:312, 1987 (#52).

[7] Fagrell, B.: Microcirculatory methods for the clinical assessment of hypertension, hypotension, and ischemia. Am. Biomed. Engr. 14:163-173, 1986.

[8] Fagrell, B.: The relationship between *macro- and microcirculation--clinical* aspects. Acta. Pharmacol. Toxicol. 58 (Supp II): 67-72, 1986.

[9] Fagrell, B., Fronek, A., Intaglietta, M.: A microscope television system for studying flow velocity in human skin capillaries. Am.J. Physiol. 233(2):H318-H321, 1977.

[10] Joshua, I.G., Wiegman, D.L., Harris, P.D., Miller, F.N.: Progressive microvascular alterations with the development of renovascular hypertension. Hypertension 6 #1:61-67, 1984.

[11] Mayrovitz, H.N., Kang, S.J., Herscovici, B., Sampsell, R.N.: Leukocyte adherence initiation in *skeletal muscle capillaries* and venules. Microvas. Res. 33:22-34, 1987.

[12] Miller, F.N., Hammerschmidt, D.E., Anderson, G.L., Moore, J.N.: Protein loss induced by complement activation during peritoneal dialysis. Kidney International, 25:480-485, 1984.

[13] Miller, F.N., Joshua, I.G., Fleming, J.T., Parekh, N.: Histamine induced protein leakage in hypertensive rats. Am. J. Physiol. 250:H284-H290, 1986.

[14] Miller, F.N., Wiegman, D.L.: Anesthesia-induced alteration of small vessel responses to norepinephrine. Eur. J. Pharmacol. 44:331-337, 1977.

[15] Richardson, D., Schmitz, H., Borchers, N.: Relative effects of static muscle contraction on digital artery and nailfold capillary blood flow velocities. Microvas. Res. 31:157-169, 1986.

[16] Wiegman, D.L., Miller, F.N., Harris, P.D.: Modification of alpha-adrenergic responses of small arteries by altered PCO_2 and pH. Europ. J. Pharmacol. 57:307-315, 1979.

AN EMPIRICAL BAYES DECISION RULE OF TWO-CLASS PATTERN RECOGNITION

Tze Fen Li and Dinesh S. Bhoj, Rutgers University at Camden

Abstract

In the pattern classification problem, it is known that the Bayes decision rule, which separates two classes of patterns gives a minimum probability of misclassification. In this study, the conditional density functions are known, but the prior probability of each class is unknown. A set of past observations (or a training set) of unknown classes is used to estimate the unknown true prior probability and hence is used to construct an empirical Bayes decision rule, which separates two classes and which can make the probability of misclassification arbitarily close to that of the Bayes rule. The results of a Monte Carlo simulation study are presented to demonstrate the favorable prior estimation and the classification performed by the empirical Bayes decision rule.

Key words and phrases: classification, empirical Bayes, pattern recognition.

1. Introduction

Essentially, there are two different approaches to solving classification problems. One approach is to find a Bayes decision rule, which separates two classes based on the present observation X and minimizes the probability of misclassification [3,7]. This approach requires sufficient information about the conditional density function $f(x \mid w)$ of X given class w and the prior probability $p(w)$ of each class, otherwise, the conditional density and the prior probability have to be estimated through a set of past observations (or a training set of sample patterns with known classes). On the other hand, if very little is known about statistical properties of the pattern classes, a discriminant function $D(X, \theta_1, \theta_2, ..., \theta_m)$ can be used. A learning automation and an algorithm are designed to of the discriminant function. After learning, this function is used to separate pattern classes [1,2,5,6]. For this approach, it is not easy to define the functional form and its parameters. Moreover, the discriminant function after learning will not be able to give the minimum probability of misclassification. In this study, the first approach is applied to solving two-class pattern problem.

The conditional density functions $f(x \mid w)$ are known, but the prior probability of each class is unknown. A set of n past observations of unknown classes is used to estimate the true unknown prior probability and this is used to construct a decision rule, called an empirical Bayes (EB) rule [4], which is used to separate two classes. It will make the probability of misclassification arbitarily close to that of the Bayes rule. The results of a Monte Carlo simulation study with one-dimensional distributions are presented to demonstrate the favorable estimation of the unknown prior probability and the boundary point of two classes made by the EB decision rule.

2. Classification Of Two Classes

Let X be the present observation which belongs to one of two classes c_1 and c_2. Consider the decision problem consisting of determining whether X belongs to c_1 or c_2. Let $f(x \mid w)$ be the conditional density function, where $w = c_1$ denotes class 1 and $w = c_2$ denotes class 2. Let θ be the prior probability of $w = c_1$. Let d be a decision rule. A simple loss function is used such that the loss is 1 when d makes a wrong decision and the loss is 0 when d makes a right decision. Let $R(\theta, d)$ denote the risk function (the probability of misclassification) of d. Let L and U be two regions separated at a point z by the decision rule d in the domain of X, i.e., d decides c_1 when $X \in L$ and decides c_2 when $X \in U$. Then

$$R(\theta, d) = \int_U \theta f(x \mid c_1)dx + \int_L (1-\theta) f(x \mid c_2)dx \tag{1}$$

Let D be the family of all decision rules which separate two classes. For θ fixed, let minimum paobability of misclassiffation be denoted by

$$R(\theta) = \inf_{d \in D} R(\theta, d) \tag{2}$$

A decision rule d_θ which satisfies (2) is called the Bayes decision rule with respect to the prior θ and given by

$$\begin{aligned} d_\theta(x) &= c_1 \quad \text{if} \quad \theta f(x \mid c_1) > (1-\theta) f(x \mid c_2) \\ &= c_2 \quad \text{otherwise} \end{aligned} \tag{3}$$

In the empirical Bayes (EB) decision problem [4], the past observations (w_m, X_m), m=1,2,...,n and the present observation X are i.i.d.. The EB decision problem is to establish a decision rule based on the set of past observations $\underline{X}_n = (X_1, ..., X_n)$. This can be constructed as using $\underline{X}_n$ to select a decision rule $t_n(\underline{X}_n)$ which determines whether the present observation X belongs to c_1 or c_2. Let $p(x_m \mid \theta)$ be the marginal density of X_m with respect to the prior distributions of classes, i.e.,

$$p(x_m \mid \theta) = \theta f(x_m \mid c_1) + (1-\theta) f(x_m \mid c_2).$$

We divide the interval [0,1] into k subintervals and a finite discrete distribution Φ is placed on $\theta \in [0,1]$ such that $\Phi(\theta = \theta_i) = \frac{1}{k}$, where θ_i is the middle point of the i-th subinterval $[\frac{i-1}{k}, \frac{i}{k}]$, $i = 1, ..., k$. Let $h(\theta_i \mid \underline{x}_n)$ be defined by

$$h(\theta_i \mid \underline{x}_n) = \frac{\prod_{m=1}^n p(x_m \mid \theta_i)}{\sum_{j=1}^k \prod_{m=1}^n p(x_m \mid \theta_j)} \qquad i = 1, ..., k \tag{4}$$

which is the conditional probability of θ_i given $\underline{X}_n = \underline{x}_n$. The conditional expectation $E[\theta \mid \underline{X}_n]$ was shown [8] to converge a.s. to a point $\hat{\theta} \in [0,1]$ with $|\hat{\theta} - \mu| < \frac{1}{k}$ with respect to the true prior probability $p(w = c_1) = \mu$. Our EB decision rule is obtained by replacing the unknown θ in (3) by $E[\theta \mid \underline{X}_n]$ and is written as

$$\begin{aligned} \hat{t}(\underline{X}_n)(X) &= c_1 \quad \text{if} \quad \frac{f(X \mid c_1)}{f(X \mid c_2)} > E[\theta \mid \underline{X}_n]^{-1} - 1 \\ &= c_2 \quad \text{otherwise} \end{aligned} \tag{5}$$

The EB decision rule (5) is used to separate two classes and the simulation results will be presented in the next section.

3. Simulation Results

In this section, we generate a set of observations (a training set) $\underline{X}_n$ which are used to estimate the true prior probability μ of class 1 and establish an empirical Bayes rules for each of three cases. Each EB rule will determine a boundary point of two classes. A normal distribution and a uniform distribution are used to be the conditional density function $f(x \mid w)$:

Class 1:	**N(0,1)**	**U(0.5)**
Class 2:	**N(2,1)**	**U(1.0)**

The prior distribution is unknown. For the normal distribution, 400,500 and 600 observations are generated from an IBM-PC microcomputer and the percentage of observations from class 1 is $\theta = 0.3,0.5$ and 0.7. The simulation results are given in Table 1. A set of 600 observations gives a satisfactory estimation $E[\theta \mid \underline{X}_n]$ of the true prior probability μ, which are 0.3003, 0.4995 and 0.6925 respectvely. The boundary points of two classes determined by the Bayes rule and the EB rule are also given in Table 2. Table 2 shows the boundary points provided by the EB rule which are close to that of the Bayes rule.

Table 1 The Estimation of $E[\theta \mid \underline{X}_n]$ Normal distribution with means=0 and 2,and equal variance=1. Uniform didtribution with means=0.5 and 1.0.

True Prior μ	No. of observations	$E[\theta \mid \underline{X}_n]$ Normal	Uniform
0.3	400	0.3002	0.3001
0.3	500	0.3002	0.3005
0.3	600	0.3003	0.3001
0.5	400	0.4680	0.4976
0.5	500	0.4959	0.4993
0.5	600	0.4995	0.4993
0.7	400	0.6204	0.6948
0.7	500	0.6662	0.6955
0.7	600	0.6925	0.6962

For the uniform distribution, 400, 500 and 600 observations are generated for θ = 0.3,0.5 and 0.7. The simulation results are also given in Table 1. The set of 600 observations tions gives a satisfactory estimation of the true prior μ, which are 0.3001, 0.4993 and 0.6962 respectively. The boundary points of two classes are given in Table 3. Table 3 shows that the boundary point determined by the EB rule is the same as that of the Bayes rule.

Table 2. The boundary points of two classes given by the Bayes rule and the EB rules for normal distributions.

True prior μ	Bayes rule	EB rule
0.3	0.5764	0.5771
0.5	1.0000	0.9990
0.7	1.4237	1.4060

Table 3. The boundary points of two classes given by the Bayes rule and the BE rules for uniform distributions.

True prior μ	Bayes rule	EB rule
0.3	0.5000	0.5000
0.5	0.5000	0.5000
	-1.0000	-1.0000
0.7	1.0000	1.0000

Note: 0.5000–1.0000 means that the boundary point can be anywhere between .5 and 1.

References

[1]. A. G. Barto and P. Anandan, "Pattern recognizing stochastic learning automata," IEEE Trans. Syst., Man, Cyber., Vol. SMC-15, PP. 360-375, May 1985.

[2]. H. Do-Tu and M. Installe, "Learning algorithms for non-parametric solutions to the minimum error classifi- cation problem," IEEE Trans. Comput., Vol. C-27, pp. 648-657, July 1978.

[3]. K. Kukunaga, Introduction to statistical pattern recognition. New York: Academic Press, 1972.

[4]. H. Robbins, "An empirical Bayes approach to statis- tics," Proc. Third Berkeley Symp. Math. Statist. prob., Vol. 1, University of California Press. pp. 157-163, 1956.

[5]. M. A. L. Thathachar and K. R. Ramakrishnan, "A coo- perative game of a pair of learning automata," Automa- tion, Vol. 20, pp. 797-801, June 1984.

[6]. M. A. L. Thathachar and P. S. Sastry, "Learning optimal discriminant functions through a cooperative game of automata," IEEE Trans. Syst., man, Cybern., Vol. SMC-17, pp.73-85, Jan. 1987.

[7]. T. Y. Young and T. W. Calvert, Classification, Estimation and Pattern Recognittion, New York: Elsevier, 1974.

[8]. T. F. Li and D. S. Bhoj, An empirical Bayes approach to pattern recognition, Department of Mathematics, Rutgers University, (1987). (submitted for publication)

STATISTICAL MODELING OF A PRIORI INFORMATION FOR IMAGE PROCESSING PROBLEMS: A Mathematical Expression of Images

Z. Liang, Duke University Medical Center

ABSTRACT

A general mathematical expression of images is presented intended to reflect the intrinsic probabilistic information of image density distribution, in terms of a priori image (or source) probability density functions. It strongly resembles the entropy form defined by Kullback and Leibler and has the defined contents of a priori source distribution probabilistic information. The expression reduces to the form of Shannon's entropy if a uniform a priori source probability distribution is assumed. A Bayesian analysis incorporating the a priori source probabilistic information is studied in treating observed data obeying Poisson statistics. A system of equations determining the Bayesian solution is given which maximizes the a posteriori probability given the observed data. A Bayesian imaging algorithm approaching to the solution iteratively is derived by employing an expectation maximization technique. Tests of the Bayesian algorithm with uniform and non-uniform a priori probabilities are carried out for computer generated ideal data and experimental phantom imaging data containing Poisson noise. Good quality images are obtained. Preliminary study of maximizing the a priori source distribution probabilistic information is also presented.

INTRODUCTION

Statistical modeling of image processing problems of ill-posed in inverse process [1] has been enhanced in recent years by use of the maximum entropy (ME) [2-3] and the maximum likelihood (ML) [4-5] analysis. Although some effort has been made to consider both the source entropy and data likelihood information [6-7], statistical modeling of the image processing problems has not yet been extensively investigated. For that purpose, a statistical model of a priori source distribution probabilistic information has been proposed intended to reflect the intrinsic probabilistic information of source distribution [8]. The model considers the statistcal behavior of individual source element and incorporates the a priori source information via maximum entropy analysis. This statistical model has now been developed to consider two general classes of a priori source probabilistic information in consistent with the random process of source (or image) density distribution: (a). uniform and (b). non-uniform a priori image probability distributions. Mathematical expressions of the statistical model of images containing the uniform and non-uniform a priori probabilistic information are formulated. The image expressions strongly resemble the entropy forms defined respectively by Shannon [9] and Kullback et al [10] and have the defined content of a priori image density probability distribution. These formulas of a priori source distribution probabilistic information imply a principle of maximum a priori probability (PMAPP), of which the Jaynes' maximum entropy principle (MEP) [11] may be a special case. A Bayesian analysis incorporating the a priori source probabilistic information, where the data likelihood (probability) function is assumed to reflect the Poisson statistics of photon detection, is studied. Other likelihood functions of uncorrelated and correlated Gaussian data are given in Appendix A. A system of equations determining the Bayesian solution is derived which maximizes the a posteriori probability given the observed data. A Bayesian imaging algorithm approaching to the solution iteratively is derived by employing, among many other iterative schemes [12-13], the expectation maximization (EM) technique [14]. As a simple example rather than using the EM technique, the steepest descent method [12] is used for the Bayesian solution as shown in Appendix B. Preliminary study of maximizing the a priori source probability using the Lagrange parameter technique [6] and the recursive Picard method [15] are given respectively in Appendices C and D. Tests of the EM Bayesian algorithm and other iterative algorithms derived in the Appendices with the uniform and non uniform a priori source probabilistic information are carried out for computer generated ideal data and experimental phantom imaging data containing the Poisson noise. Good quality images are obtained. A filtering criterion function is used to quantitatively indicate the convergence performance of the iterative Bayesian algorithm.

A PRIORI INFORMATION FUNCTIONS

The source distribution region is, as usual in digital image processing, divided into J source elements (or voxels). Each voxel has an average value over its volume, $\{\phi_j\}$, $j=1, 2, \ldots, J$. In nuclear isotope imaging, ϕ_j stands for the Gamma photon emission from voxel j per unit time at time=0; in X-ray imaging, it represents the attenuation density of voxel j; in optical picture processing, it is the radiance value of voxel j; in scanning electron microscopes, it is the transmittance of voxel j; and in NMR imaging, it reflects the intensity of voxel j in the spectrum space. In the following sections, ϕ_j is referred to generally as the strength or density of voxel j.

If the source strengths $\{\phi_j\}$ are hypothetically quantized into strength units (or photon "balls"), then ϕ_j represents the number of the strength units, or the photon balls. If the total number of strength units $N = \sum_j \phi_j$ can be assumed to be fixed, the source strength distribution can then be characterized as a random process in which the N strength units distribute randomly over the J voxels. Let $p_j(\phi_j)$ represent the a priori probability of a strength unit falling into voxel j, the a priori source distribution probability is then expressed as [8]:

$$P(\Phi) = \frac{N!}{\prod_j \phi_j!} \prod_{j=1}^{J} [\, p_j(\phi_j)\,]^{\phi_j} . \qquad (1)$$

The $P(\Phi)$ of function (1) is the a priori source probability function. It reflects a statistical random process of image density distribution considering a priori probability distribution information $\{p_j(\phi_j)\}$. The underlying assumptions of function (1) are very general and function (1) can be applicable in many image processing problems.

By the definition of probability (i.e., with total N density units, there are ϕ_j units falling into voxel j):

$$p_j(\phi_j) = \lim_{N \to \infty} \frac{\phi_j}{N} ,$$

it is assumed that the a priori probability may be approximated as:

$$p_j(\phi_j) = \frac{\bar{\phi}_j}{N} \qquad (2)$$

where $\bar{\phi}_j$ represents the a priori mean strength of voxel j. The larger the value N the higher the information content of data statistics, and so the closer ϕ_j will approach to $\bar{\phi}_j$. The estimation of $\bar{\phi}_j$ is quite important for the optimal solution of $\{\phi_j\}$ given the observed data. This will be discussed later.

If a maximum a priori probability principle applies, then maximizing the probability function $P(\Phi)$ is equivalent to maximizing the log function:

$$H(\Phi) \equiv \ln P(\Phi) = \ln(N!) + \sum_j [\, \phi_j \ln p_j(\phi_j) - \ln(\phi_j!)\,]$$

$$= \ln(N!) + \sum_j [\, \phi_j \ln(\bar{\phi}_j) - \phi_j \ln(N) - \ln(\phi_j!)\,] . \qquad (3)$$

Using the Stirling's formula

$$\ln(N!) \approx N \ln(N) - N$$

and the contraint of $N = \sum_j \phi_j$, function (3) becomes:

$$H(\Phi) = -\sum_j \phi_j \ln\left(\frac{\phi_j}{\bar{\phi}_j}\right). \quad (4)$$

Function (4) is the general mathematical expression of images containing the a priori image density distribution probabilistic information under the principle of maximum a priori probability. It strongly resembles the entropy form defined by Kullback and Leibler [10] and has the defined contents of image density $\{\phi_j\}$ and the a priori mean information $\{\bar{\phi}_j\}$.

If the a priori probability distribution $\{p_j(\phi_j)\}$ is uniform, i.e., $\bar{\phi}_j = \bar{\phi} = N/J$, function (4) reduces to:

$$H(\Phi) = -\sum_j \phi_j \ln(\phi_j) + N \ln(N/J). \quad (5)$$

Under the PMAPP, function (5) strongly resembles the entropy form defined by Shannon [9] plus a constant. It has the defined contents of image density $\{\phi_j\}$ and implies the assumption of uniform a priori probability distribution. An image information analysis can be carried out in a similar way as Shannon's communication analysis [9]. An $m \times n$ image can be represented by a point in $m \times n$ multidimensional space. The mapping from a point of image in the $m \times n$ image space to multidimensional data space can be ideally assumed as one to one mapping. The distortion and noise contamination in the measurement process blure the point in the data space to be a small region. The inverse mapping, therefore, produces more than one point (or a region) in the image space. Different objective criteria impose different constraints on the inverse mapping and on the selection of the corresponding point in the image region of $m \times n$ dimensions. Function (1) reflects then the probability distribution of the images over the image region. Under the principle of maximum a priori probability, function (4) can be a measure of the closeness of $\{\phi_j\}$ to $\{\bar{\phi}_j\}$ and function (5) the measure of the probable distribution configurations of $\{\phi_j\}$ over the image region in the $m \times n$ dimensional space. The most likely point in the image region of $m \times n$ dimensions is the most likely distribution of the image density $\{\phi_j\}$ over the $m \times n$ voxels. In image processing, an image having density distribution $\{a\ \phi_j\}$ may not be distinct from the image having density $\{b\ \phi_j\}$ (where a and $b \neq a$ are constants). In this case, the images are said to be degenerate. The degeneracy is reflected, in the $m \times n$ multidimensional space, as a line passing through the origion of coordinates.

As commonly assumed in most imaging applications, the mapping is linear and can be expressed as [16]:

$$Y_i = \sum_j R_{ij} \phi_j + e_i, \quad i = 1, 2, \ldots, I \quad (6)$$

where $\{Y_i\}$ are the data elements and can be represented as a point in the I-dimensional data space; e_i is the noise component and R_{ij} the probability of receiving an image density unit from voxel j for measurement point i (or projection ray Y_i). In image restoration application, R_{ij} is the point spread function (PSF) of the imaging system [16].

Since the noise components $\{e_i\}$ are unpredictable, the statistical property of data fluctuation would be considered in terms of probability distribution. If each data element Y_i obeys Poisson statistics around the mean $\sum_j R_{ij}\phi_j$ and all the data elements $\{Y_i\}$ are uncorrelated with each other, the data probability distribution is [4,17-18]:

$$P(\mathbf{Y} \mid \Phi) = \prod_{i=1}^{I} exp\left(-\sum_j R_{ij}\phi_j\right)\left(\sum_j R_{ij}\phi_j\right)^{Y_i} / Y_i!. \quad (7)$$

The $P(\mathbf{Y} \mid \Phi)$ of function (7) is the data probability function. It reflects the Poisson nature and uncorrelated fluctuations of measurements. It is noted that the means of $\{\sum_j R_{ij}\phi_j\}$ are, of course, correlated with each other.

Other probability distributions of data vector $\mathbf{Y}$ have been considered in the previous work [18] and are, as references, given in Appendix A.

The likelihood function of the data distribution is expressed as [17-19]:

$$L(\mathbf{Y} \mid \Phi) \equiv lnP(\mathbf{Y} \mid \Phi)$$

$$= \sum_i \left[-\sum_j R_{ij}\phi_j + Y_i \ln\left(\sum_j R_{ij}\phi_j\right) - \ln(Y_i!)\right]. \quad (8)$$

Function (8) is a measure of the likelihood of which the source distribution $\{\phi_j\}$ would have given rise to the observed data obeying the Poisson statistics.

A Bayesian analysis providing the maximum a posteriori solution is studied in the following section. It considers both the likelihood character of data fluctuations (8) and the a priori source probabilistic information (4). A discussion on considering the a priori source information (4) and the linear data constraints of Eqs.(6) is given in Appendix C, where the Lagrange parameter technique [6] is employed.

BAYESIAN ANALYSIS AND ALGORITHM

Bayesian analysis provides a maximum a posteriori solution Φ^* which considers both the data statistics and the a priori source distribution probabilistic information. From Bayes' Law:

$$P(\Phi \mid \mathbf{Y}) = P(\mathbf{Y} \mid \Phi)\, P(\Phi) / P(\mathbf{Y}), \quad (9)$$

the Bayesian function is given by:

$$g(\Phi) \equiv lnP(\Phi \mid \mathbf{Y}) = lnP(\mathbf{Y} \mid \Phi) + lnP(\Phi) - lnP(\mathbf{Y}). \quad (10)$$

Considering the Poisson nature of data fluctuations (8) and the a priori probabilistic information of source distribution (4), the Bayesian function is:

$$\begin{aligned} g(\Phi) &= L(\mathbf{Y} \mid \Phi) + H(\Phi) - lnP(\mathbf{Y}) \\ &= \sum_i \left[-\sum_j R_{ij}\phi_j + Y_i \ln\left(\sum_j R_{ij}\phi_j\right)\right] \\ &\quad - \sum_j \left[\phi_j \ln(\phi_j / \bar{\phi}_j)\right] + C(\mathbf{Y}) \end{aligned} \quad (11)$$

where $C(\mathbf{Y}) = -lnP(\mathbf{Y}) - \sum_i \ln(Y_i!)$ is independent of Φ. Since $C(\mathbf{Y})$ does not effect the determination of the Bayesian solution Φ^* which maximizes function $g(\Phi)$, it will be omitted later.

A system of equations determining the Bayesian solution Φ^* is derived by maximizing the Bayesian function $g(\Phi)$,

$$\left.\frac{\partial g(\Phi)}{\partial \phi_i}\right|_{\Phi=\Phi^*} = 0$$

i.e.,

$$\sum_i R_{ii}\left(Y_i / \sum_j R_{ij}\phi_j^*\right) - \sum_i R_{ii} = \ln(\phi_i^*) - \ln(\bar{\phi}_i) + 1. \quad (12)$$

Since $g(\Phi)$ is strictly concave, i.e., for any non-vanishing vector Z, there is:

$$\begin{aligned} Z^T[\nabla^2 g(\Phi)]Z &= -\left[\sum_i Y_i\left(\sum_j R_{ij}Z_j / \sum_j R_{ij}\phi_j\right)^2 \right. \\ &\quad \left. + \sum_j Z_j^2/\phi_j\right] < 0, \end{aligned}$$

the solution Φ^* is uniquely determined by the Eqs.(12).

A Bayesian algorithm carrying out the calculation of the solution Φ^* iteratively is derived by employing, among many other iterative schemes [12-13], the EM technique [4,8,14,19]:

$$\phi_i^{(n+1)} = \phi_i^{(n)} \frac{\sum_i R_{ii}\left(Y_i / \sum_j R_{ij}\phi_j^{(n)}\right)}{\sum_i R_{ii} + \xi_i^{(n)} Z_i^{(n)}} \quad (13)$$

and

$$Z_i^{(n)} = -\left.\frac{\partial H(\Phi)}{\partial \phi_i}\right|_{\phi_k = \phi_k^{(n)} + \epsilon\,\delta_k^{(n)}} \quad (14)$$

where $\epsilon \approx 1$ and $\delta_i^{(n)} = \phi_i^{(n)} - \phi_i^{(n-1)}$ are assumed for easy computation, and $\xi_i^{(n)}$ is an adjustable sigmoidal parameter chosen to gradually impose the effect of the a priori information $H(\Phi)$:

$$\xi_i^{(n)} = \frac{A\, n^{\tau}}{B + n^{\tau}} \sum_i R_{ii} \quad (15)$$

with A, B, and τ constant.

Note that while the approximation of $\phi_k = \phi_k^{(n)} + \epsilon\, \delta_k^{(n)}$ in Eq.(14) is assumed for easy computation, other approximations can be used. The gradually increase of $\xi_k^{(n)}$ as a function of iterative index n is quite important for optimal results but the values of A, B, and τ can vary significantly from those used in this paper.

(a). For the uniform a priori probabilistic information (5):

$$Z_k^{(n)} = ln\,(\phi_k^{(n)} + \epsilon\, \delta_k^{(n)}) + 1\,; \qquad (16)$$

(b). For the non-uniform a priori information (4):

$$Z_k^{(n)} = ln\,(\phi_k^{(n)} + \epsilon\, \delta_k^{(n)}) - ln\,(\bar{\phi}_k) + 1\,. \qquad (17)$$

The Bayesian algorithm of Eq.(13) considering the uniform and non-uniform a priori information (16) and (17) respectively will be applied to computer generated ideal data and experimental phantom imaging data containing Poisson noise in the following section. A discussion on maximizing $g(\Phi)$ via the steepest descent technique [12] will be, as a simple example, given in Appendix B. The discussions on emphasizing the a priori $P(\Phi)$ are given in Appendices C and D respectively by use of the data constraints (6) and (8).

RESULTS

The BIP algorithm of Eq.(13) considering (16) and (17) respectively is tested in two different imaging situations: (a). computer generated noise-free data, where one dimensional image restoration and two dimensional image restoration and reconstruction are considered; and (b). computer generated and experimental phantom imaging data containing Poisson noise, where the similar tests as in (a) are carried out. To facilitate the calculation of the algorithm and the test function mentioned below, only one dimensional results of convergence performance of the algorithm are reported. Multidimensional calculation is straightforward. Preliminary results using the iterative algorithms derived in the Appendices are also reported.

(i). one dimensional image restoration results

In the case of noise-free data, the actual source distribution $\{S_j\}$ consists of two point sources of 57 strength units each, separated by 8 voxel units, superimposed upon a uniform background of 3 strength units, as shown in Fig.1 by the solid line. The ideal data (noise-free) $\{Y_i\}$ are calculated from $Y_i = \sum_j R_{ij} S_j$, as shown in Fig.1 by the dotted line, where the functional form of $\{R_{ij}\}$ is assumed as:

$$R_{ij} \approx e^{-ln(2)(i-j)^2/T^2} \qquad (18)$$

with $T = FWHM/2 = 4.5$ voxel units, as shown in Fig.1 by the broken line. It reflects a quite poor spatial resolution imaging system.

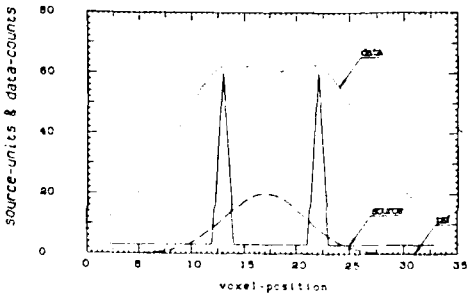


Fig.1 Source distribution (solid line), noise-free data (dotted line) and point spread function (broken line).

Fig.2 compares the results of the a priori uniform BIP algorithm (u) of Eqs.(13) and (16) (dotted line) and the a priori non-uniform BIP algorithm (n) of Eqs.(13) and (17) (solid line) after 50 iterations for the ideal data. The initial estimate $\Phi^{(0)}$ and the mean values $\{\bar{\phi}_j\}$ are chosen as:

$$\phi_j^{(0)} = Y_j / \sum_i R_{ij}\,, \text{ and } \bar{\phi}_j = \phi_j^{(n)} + \eta\, \delta_j^{(n)} \qquad (19)$$

where $\eta \geq \epsilon$ is a constant. The values of $A = 1$, $B = 100$, $\tau = 1$ and $\eta = 5$ are chosen.

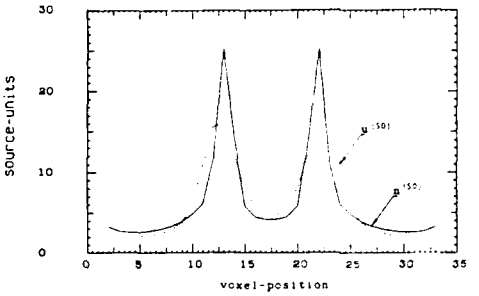


Fig.2 Comparison of the a priori uniform (u, dotted line) and non-uniform (n, solid line) BIP algorithms after 50 iterations for the noise-free data.

In order to quantitatively evaluate the performance of the a priori uniform and non-uniform BIP algorithms, a test function of root-mean-square criterion is used,

$$\psi_0 = [\, \sum_j (\phi_j^{(n)} - S_j)^2 / \sum_j (S_j - \bar{S})^2 \,]^{\frac{1}{2}} \qquad (20)$$

where $\bar{S}$ is the mean of $\{S_j\}$.

Fig.3 shows the results of ψ_0 as a function of iterative index n. Since the neighboring voxels around the two point sources play a dominant role in the results of the test function, a smoothing weight filtering is applied before using the test function. In another words, the test function is modifed as [13]:

$$\psi_1 = [\, \sum_j (\bar{\phi}_j^{(n)} - \bar{S}_j)^2 / \sum_j (\bar{S}_j - \bar{S})^2 \,]^{\frac{1}{2}} \qquad (21)$$

where the weighting process is expressed as:

$$\bar{S}_j = \sum_i w_{ij} S_i \;/\; \sum_i w_{ij}\,, \; i = j-2,\; j-1,\; j,\; j+1,\; j+2 \qquad (22)$$

and

$$w_{j-2,j} = 0.2\,, \quad w_{j-1,j} = 0.5 \qquad (23)$$
$$w_{j,j} = 1.0\,, \quad w_{j+1,j} = 0.5\,, \quad w_{j+2,j} = 0.2$$

and similarly for $\bar{\phi}_j^{(n)}$.

The results of ψ_1 are shown in Fig.4. The modified test function reflects more accurately the performance of convergence of the iterative algorithms since it considers both the amplitude and spatial components. As shown later, such improvement with the modified test function is more significant when the data is noisy.

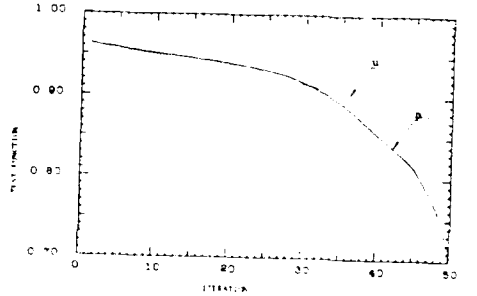


Fig.3 Results using test function (20) for the BIP algorithms in the case of noise-free data.

In the case of experimental imaging data, a simple one dimensional equivalent phantom is prepared by threading two parallel catheters (separated by about 7 voxel units) containing a solution Co^{57} through a stainless steel screen. Two dimensional data is obtained by imaging the phantom using a Picker Dyna Camera Model No.4 without collimator and is arranged as a 32×32 matrix, with the two lines of tubing oriented in the colume direction as shown in Fig.13. Row 16 of the data matrix is indicated in Fig.5 by the stars and is used as the one dimensional imaging data $\{Y_i\}$. Neglecting the effect of the finite length of the tubing, $\{Y_i\}$ can be viewed as imaging data

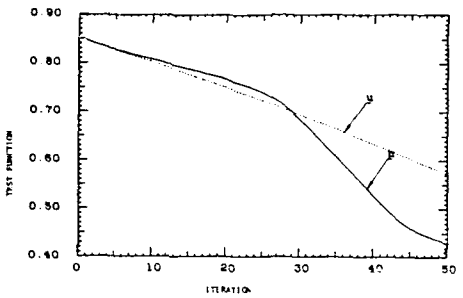


Fig.4 Results using test function (21) in the noise-free case.

from a double element source distribution $\{S_j\}$ (the projection of the two line sources along the parallel direction superimposed on a uniform background) in a one dimensional geometry, where the means are $\{\sum_j R_{ij} S_j\}$.

$\{R_{ij}\}$ is obtained by imaging a point source of Co^{57} at the same depth as the phantom (as shown in Fig.14) and is formed as a matrix using the technique [16]. One row of $\{R_{ij}\}$ is shown in Fig.5 by the solid line in which the center value is normalized to 40.

Fig.6 compares the results of the BIP algorithm with the a priori uniform (dotted line) and non-uniform (solid line) information after 50 iterations for the experimental imaging data containing Poisson noise.

The convergence performances of the BIP algorithms using the test function (20) are shown by Fig.7. The results using the modified test function (21) are shown by Fig.8.

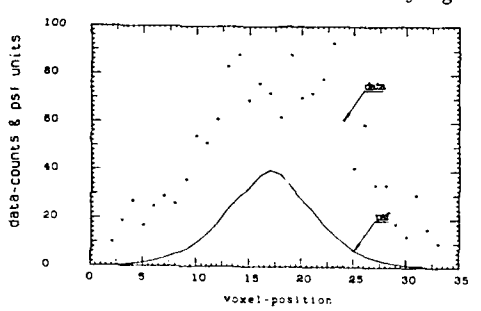


Fig.5 One dimensional experimental phantom imaging data (stars) and point spread function (solid line).

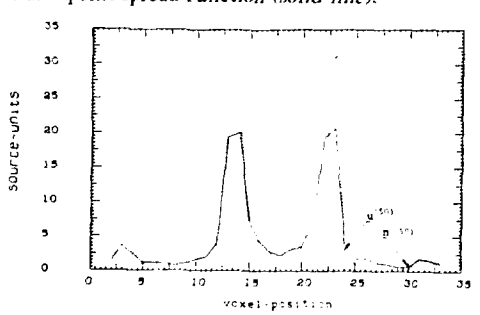


Fig.6 Comparison of the a priori uniform (u, dotted line) and non-uniform (n, solid line) BIP algorithms after 50 iterations for the experimental phantom imaging data containing Poisson noise.

(ii). two dimensional image restoration results

In the case of noise-free data, the actual source distribution $\{S_j\}$ is shown by Fig.9. It consists of two point sources of 109 strength units each, separated by 8 voxel units, superimposed upon a uniform background of 1 strength unit. A two dimensional PSF of Eq.(18) is assumed. Fig.10 shows the noise-free data distribution calculated from the convolution of $\{\sum_j R_{ij} S_j\}$. Fig.11 shows the result using the a priori uniform BIP algorithm after 100 iterations for the noise-free data.

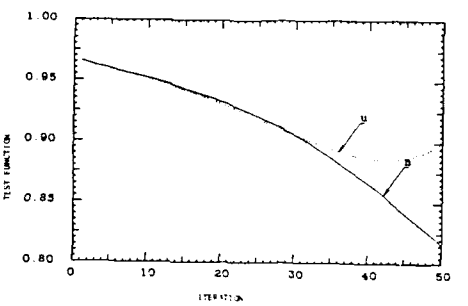


Fig.7 Results using test function (20) for the BIP algorithms in the case of experimental phantom imaging noisy data.

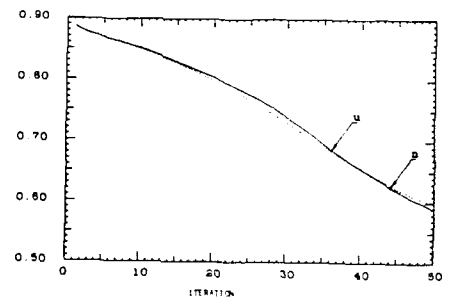


Fig.8 Results using test function (21) in the case of noisy data.

The result of the a priori non-uniform BIP algorithm for the noise-free data after 100 iterations is shown in Fig.12.

For the experimental phantom imaging tests, Figs.13 and 14 show respectively, as mentioned before, the experimental imaging noisy data $\{Y_i\}$ from a phantom containing two parallel lines of tubing and the point spread function of the camera system from which $\{R_{ij}\}$ is formed [16].

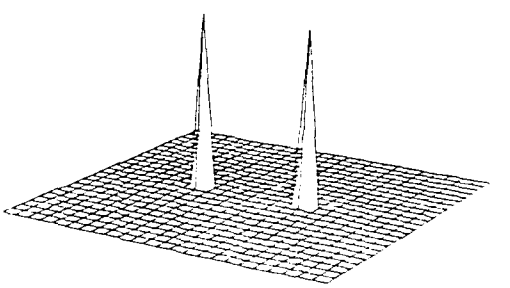


Fig.9 Two dimensional source distribution consisting of two point sources, superimposed upon a uniform background.

fig.10

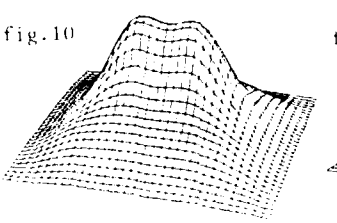


fig.11

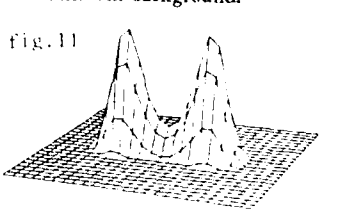


Fig.10 Two dimensional noise-free data distribution, calculated from the convolution of the source distribution of Fig.9 and a two dimensional PSF of Eq.(18).

Fig.11 Result of the a priori uniform BIP algorithm after 100 iterations for the noise-free data.

Fig.15 shows the result of the a priori uniform BIP algorithm after 25 iterations for the phantom imaging data. The result of the a priori non-uniform BIP algorithm after 25 iterations for the noisy data is shown by Fig.16.

fig.12

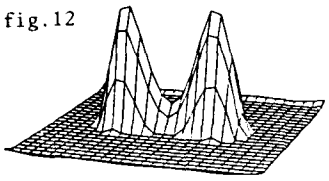


fig.13

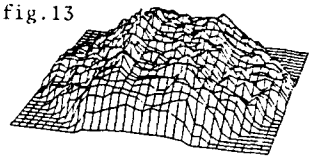


Fig.12 Result of the a priori non-uniform BIP algorithm after 100 iterations for the noise-free data.

Fig.13 Two dimensional experimental phantom imaging data.

fig.14

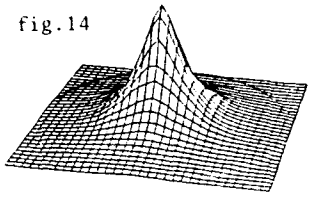


fig.15

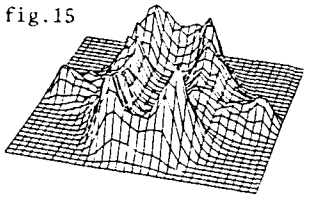


Fig.14 Two dimensional experimental point spread function.

Fig.15 Result of the a priori uniform BIP algorithm after 25 iterations for the experimental phantom imaging data containing Poisson noise.

fig.16

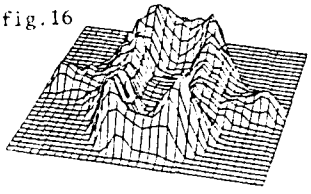


fig.17

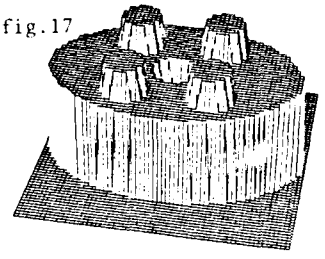


Fig.16 Result of the a priori non-uniform BIP algorithm after 25 iterations for the experimental phantom imaging noisy data.

Fig.17 An elliptical phantom containing four hot spots and a cold spot, superimposed upon a uniform background.

(iii). two dimensional image reconstruction results

Fig.17 shows the actual source distribution $\{S_j\}$ consisting of four hot spots of 4 strength units each and a cold spot of 2 strength units, superimposed upon a uniform background of 3 strength units. Outside the elliptical region, the density is zero. The rectangular region is divided into 64× 64 voxels (or pixels in two dimensions).

The projection rays are calculated from $\{\sum_j R_{ij} S_j\}$ (i.e., *noise-free projections*) for a parallel beam geometry using 64 equal projection angles in the interval [0, 180] degrees, where R_{ij} is the intersection length of projection ray i and voxel j. Each projection contains 64 equally spaced projection rays. Each of the noise-free projection ray, say ray i $(\sum_j R_{ij} S_j)$, is input to a Poisson random number generator [20]. The generated Poisson randomized projection ray Y_i then has the mean $\sum_j R_{ij} S_j$. Since those Poisson randomized projection rays $\{Y_i\}$ with zero mean are set to zero, the Poisson randomized projection rays with non-vanishing means are in the range from 1 to 120 counts. The summation of the noise-free projection rays is 356341.94 and the total counts of the Poisson randomized projections is 356946.

Figs.18 and 19 show respectively the results of the a priori uniform and non-uniform BIP algorithms after 10 iterations for the noise-free projections. The results obtained by applying the BIP algorithms to the Poisson randomized projections after 10 iterations are shown by Figs.20 and 21 respectively.

Preliminary results of the iterative algorithms derived in Appendices B, C and D are shown in the following:

Fig.22 compares the results of the descent algorithm (Eqs.(B.2), (B.3) and (B.4)) with the a priori uniform (u, dotted line) and non-uniform (n, solid line) information after 25 iterations for the noise-free data shown in Fig.1.

Fig.23 compares the results of the descent algorithm used for Fig.22 after 50 iterations for the noise-free data.

fig.18

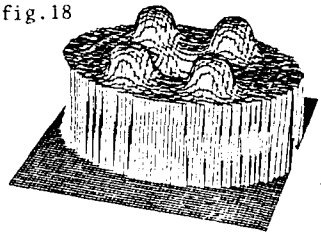


fig.19

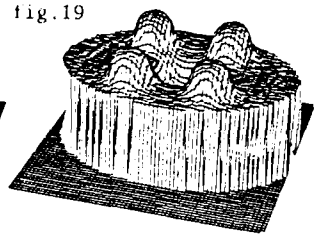


Fig.18 Result of the a priori uniform BIP algorithm after 10 iterations for noise-free projections.

Fig.19 Result of the a priori non-uniform BIP algorithm after 10 iterations for noise-free projections.

fig.20

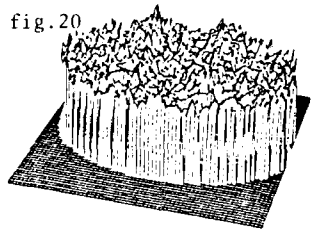


fig.21

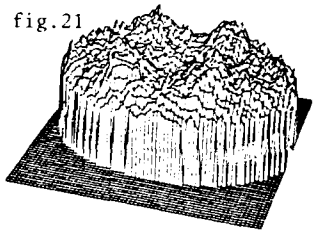


Fig.20 Result of the a priori uniform BIP algorithm after 10 iterations for Poisson randomized projections.

Fig.21 Result of the a priori non-uniform BIP algorithm after 10 iterations for Poisson randomized projections.

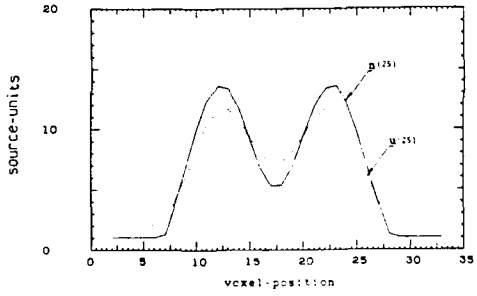


Fig.22 Comparison of the a priori uniform (u, dotted line) and non-uniform (n, solid line) descent algorithms after 25 iterations for the noise-free data shown in Fig.1.

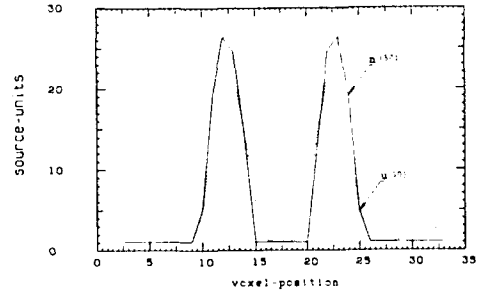


Fig.23 Comparison of the a priori uniform (u, dotted line) and non-uniform (n, solid line) descent algorithms after 50 iterations for the noise-free data.

In computer implementation of the Lagrange algorithm of Eqs.(C.7)-(C.9), uneven convergence performance is observed. The dotted lines in Figs.24 and 25 show the results of the a priori uniform Lagrange algorithm after 12 and 13 iterations respectively for the noise-free data shown in Fig.1. After the 13th iteration, no improvement is obtained. The solid lines in Figs.24 and 25 show the results of the a priori non-uniform Lagrange algorithm of Eqs.(C.4)-(C.6) after 15 and 25 iterations

respectively for the noise-free data. Smooth convergence performance with the a priori non-uniform Lagrange algorithm is observed.

Fig.26 shows the results of the a priori non-uniform Picard algorithm of Eq.(D.2) after 10 (dotted line), 25 (broken line) and 50 (solid line) iterations for the noise-free data shown in Fig.1. Since the a priori uniform information (5) is emphasized in the a priori uniform Picard algorithm of Eq.(D.3), relative flat solutions are obtained, as expected, in which the two point sources in Fig.26 are no longer resolved.

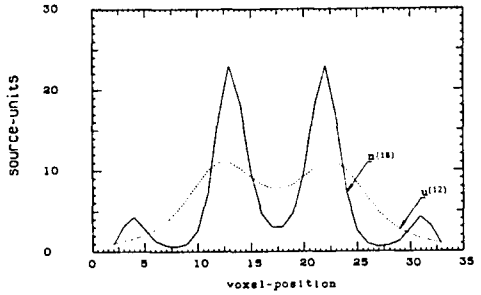


Fig.24 Comparison of the a priori uniform (u) Lagrange algorithm after 12 iterations (dotted line) and the non-uniform (n) Lagrange algorithm after 18 iterations (solid line) for the noise-free data.

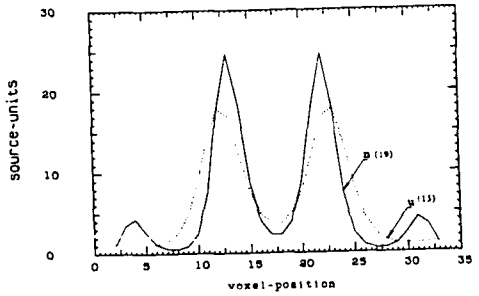


Fig.25 Comparison of the a priori uniform (u) Lagrange algorithm after 13 iterations (dotted line) and the non-uniform (n) Lagrange algorithm after 19 iterations (solid line) for the noise-free data.

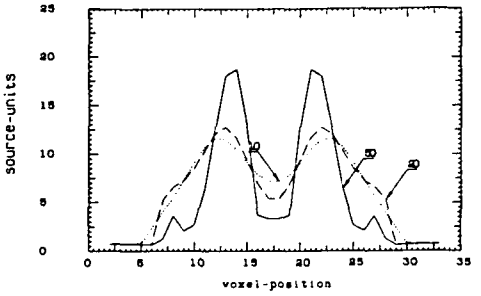


Fig.26 Results using the a priori non-uniform Picard algorithm after 10 (dotted line), 20 (broken line) and 50 (solid line) iterations for the noise-free data.

DISCUSSION

This paper presents a statistical image model intending to reflect the intrinsic probabilistic information of image density distribution. The image model is formed based on the very general assumptions of the discretization of image density units and the random distribution process of the image units over the voxels. Under the assumptions, the intrinsic probabilistic information of image density distribution is expressed mathematically as function (1). The probabilistic information function (1) can be treated either as an additional a priori source information to supplement the data likelihood solution or as a maximum criterion considering the constraints of data measurements. Incorporating additional a priori source information into the Bayesian image processing (BIP) formalism for a solution of maximum a posteriori probability has been discussed previously [18,13,8], as well as the Bayesian algorithm section and Appendix B in this paper. Maximizing a priori source probabilistic information in treating data measurements as constraints implies the principle of maximum a priori probability as mentioned in the introductory section of this paper. Preliminary study on the maximum a priori source probability subject to data constraints has been reported in references [21,13], as well as the Appendices C and D in this paper.

Under the PMAPP, the probabilistic information function (1) may be defined as a measure of the image information content if the assumptions are applicable. This can be easely seen in the following simple examples:

(a). maximizing function (1) under the assumption of uniform a priori source probability distribution (i.e., function (5)) without data constraints (or all measurements are identical) results in a uniform image;

(b). maximizing function (1) with the non-uniform a priori source probability distribution without data constraints produces a non-uniform image having density distribution proportional to the a priori mean values $\{\bar{\phi}_j\}$.

These examples are in consistence with the assumptions of the statistical image model function (1) and reflect two extreme cases of the minimal and maximal a priori information content.

The examples can be more clearly understood in the multidimensional image space. A $m \times n$ dimensional image is represented by a point in a $m \times n$ dimensional space. Example (a) specifies a point on the diagonal line (if images $\{a\ \phi_j\}$ are indistinguishable from image $\{\phi_j\}$ with any constant a, or degeneracy exists) in the multidimensional space. Example (b) produces a point on the line defined by the vector Φ. How far an image point can be brought away from the site on the diagonal line and aproaching to the point $\{\bar{\phi}_j\}$ on the line Φ depends on the data measurements. The distoration and inevitable noise in data measurements prevent the image point from reaching the point $\{\bar{\phi}_j\}$. The distoration can be removed in image processing by a suitable algorithm. However, the noise effects can not be removed in the image processing. It defines a disk range in the plane prependicular to the line Φ and passing through the point $\{\bar{\phi}_j\}$. Normalization of processed images in an iterative image processing may be helpful for convergence to the disk range. A well-conditioned algorithm considering the noise property may produce an image point within the disk range. For that propose of considering both the data statistics and a priori source information, the BIP formalism [18] has been developed to consider the pattern source information [22-23], the source distribution boundary condition [24] and the image enhancement from estimated a priori information [25].

It is noted that the information functions (4) and (5) reflect two extreme cases. Other information content between them has been investigating [8]. The application of the PMAPP to the a priori source information [8,18,22-25] is staightforward.

APPENDIX A. The Likelihood Functions of Gaussian Data

If each data element Y_i obeys Gaussian statistics around the mean value $\sum_j R_{ij}\phi_j$ and all the data elements $\{Y_i\}$ are uncorrelated, then the data probability distribution is:

$$P(\mathbf{Y} \mid \Phi) = \prod_{i=1}^{I} (2\pi\sigma_i^2)^{-1/2} exp\,[-(Y_i - \sum_j R_{ij}\phi_j)^2/(2\sigma_i^2)] \quad (A.1)$$

and the likelihood function is:

$$L(\mathbf{Y} \mid \Phi) = \sum_i [-(Y_i - \sum_j R_{ij}\phi_j)^2 / (2\sigma_i^2) - \frac{1}{2} ln\,(2\pi\sigma_i^2)]. \quad (A.2)$$

If the data elements $\{Y_i\}$ are correlated with correlation parameters $\{\chi_{il}\}$, function (A.1) becomes:

$$P(\mathbf{Y} \mid \Phi) = \prod_{il} C_{il}(\chi_{il}, \sigma_i, \sigma_l) \times \quad (A.3)$$

$$exp\,[\,-\frac{\chi_{il}}{2\sigma_i\,\sigma_l}\,(Y_i\,-\sum_j R_{i\,j}\,\phi_j\,)\,(Y_l\,-\sum_j R_{l\,j}\,\phi_j\,)\,]$$

and (A.2) becomes:

$$L\,(\mathbf{Y}\mid\Phi) = \sum_{il}[\,-\frac{\chi_{il}}{2\sigma_i\,\sigma_l}\,(Y_i\,-\sum_j R_{i\,j}\,\phi_j\,)\,(Y_l\,-\sum_j R_{l\,j}\,\phi_j\,) - ln C_{il}\,(.)\,]\,. \qquad (A.4)$$

Most regularization techniques [26-29] can be derived from the Bayesian analysis considering the data likelihood function (A.2) or (A.4) and the generic a priori source information [13,18].

APPENDIX B. Consideration of Steepest Descent Technique

Maximizing the a posteriori probability $P\,(\Phi\mid\mathbf{Y})$ by use of the steepest descent technique [12] is expressed as:

$$g\,(\Phi) = L\,(\mathbf{Y}\mid\Phi) + H\,(\Phi) - \ln P\,(\mathbf{Y}) \qquad (B.1)$$

$$= \sum_i[\,-\sum_j R_{i\,j}\,\phi_j\,+ Y_i\,\ln\,(\sum_j R_{i\,j}\,\phi_j\,)\,] - \sum_j \phi_j\,[\,\ln\,(\phi_j\,/\,\bar{\phi}_j\,)\,] + C\,(\mathbf{Y}) = \text{maximum}\,.$$

Since $g\,(\Phi)$ is strictly concave, the iterative steepest descent scheme is then given by:

$$\phi_k^{(n+1)} = \phi_k^{(n)} + \alpha\,d_k^{(n)} \qquad (B.2)$$

$$d_k^{(n)} = \nabla_k\,g\,(\Phi) \qquad (B.3)$$

$$= \sum_i R_{ik}\,(Y_i\,/\sum_j R_{i\,j}\,\phi_j^{(n)} - 1) - \ln\,(\phi_k^{(n)}\,/\,\bar{\phi}_k\,) - 1$$

and

$$\alpha = \left|\,\frac{\sum_i[\,(Y_i\,-\sum_j R_{i\,j}\,\phi_j^{(n)})\,(\sum_j R_{i\,j}\,d_j^{(n)}\,/\sum_j R_{i\,j}\,\phi_j^{(n)})\,]\,-\sum_j d_j^{(n)}\,[\,\ln\,(\phi_j^{(n)}\,/\,\bar{\phi}_j\,)+1\,]}{\sum_i[\,(\sum_j R_{i\,j}\,d_j^{(n)})^2\,/\sum_j R_{i\,j}\,\phi_j^{(n)}\,]+\sum_j[\,(d_j^{(n)})^2\,/\,\phi_j^{(n)}\,]}\,\right| \qquad (B.4)$$

For the special case of uniform a priori probabilistic information (5), the iterative scheme is given by Eqs.(B.2)-(B.4) with replacement of $\bar{\phi}_j$ by 1.

APPENDIX C. Consideration of Lagrange Parameter Technique

Maximizing the a priori probability information (4) via Lagrange parameters $\{\lambda_i\}$ is expressed by [6,30]

$$\psi(\Phi) = H\,(\Phi) - \sum_i \lambda_i\,(Y_i - \sum_j R_{i\,j}\,\phi_j\,) = \text{maximum}\,. \qquad (C.1)$$

$$= -\sum_j \phi_j\,\ln\,(\phi_j\,/\,\bar{\phi}_j\,) - \sum_i \lambda_i\,(Y_i - \sum_j R_{i\,j}\,\phi_j\,)$$

The solution for maximizing $\psi(\Phi)$ is given by:

$$\phi_k = \frac{\bar{\phi}_k}{e}\,exp\,(\sum_i R_{ik}\,\lambda_i\,)\,,\quad k = 1,\,2,\,\ldots,\,J\,. \qquad (C.2)$$

Substitute Eq.(C.2) into Eq.(C.1), there is:

$$\psi(\lambda) = \frac{1}{e}\sum_j \bar{\phi}_j\,exp\,(\sum_i R_{i\,j}\,\lambda_i\,) - \sum_i \lambda_i\,Y_i\,. \qquad (C.3)$$

Since $\psi(\Phi)$ is strictly convex, the descent approach of $\{\lambda_i\}$ iteratively is expressed as:

$$\lambda_k^{(n+1)} = \lambda_k^{(n)} + \alpha\,d_k^{(n)} \qquad (C.4)$$

$$d_k^{(n)} = -\nabla_k\,\psi(\Phi) = Y_k - \frac{1}{e}\sum_j R_{k\,j}\,\bar{\phi}_j\,exp\,(\sum_i R_{i\,j}\,\lambda_i^{(n)}) \qquad (C.5)$$

and

$$\alpha = \left|\,\frac{Z\,-\sum_j \bar{\phi}_j\,(\sum_i R_{i\,j}\,d_i^{(n)})\,exp\,(\sum_i R_{i\,j}\,\lambda_i^{(n)})}{\sum_j \bar{\phi}_j\,(\sum_i R_{i\,j}\,d_i^{(n)})^2\,exp\,(\sum_i R_{i\,j}\,\lambda_i^{(n)})}\,\right| \qquad (C.6)$$

where $Z = e\sum_i Y_i\,d_i^{(n)}$. After $\{\lambda_i\}$ have been solved, then Eq.(C.2) gives the solution Φ.

If the uniform a priori probabilistic information (5) is considered as a special case, the solution $\{\phi_j\}$ of Eqs.(C.2), (C.5) and (C.6) with replacement of $\bar{\phi}_j$ by 1.

APPENDIX D. Consideration of Recursive Picard Technique

The maximum a posteriori probability solution Φ^* is given by the system of equations (12), or

$$\phi_k^* = \bar{\phi}_k\,exp\,[\,\sum_i R_{ik}\,(\,Y_i\,/\sum_j R_{i\,j}\,\phi_j^*\,-1\,)-1\,]\,. \qquad (D.1)$$

Introducing an adjustable parameter μ for the data constaints and using the recursive Picard technique [15], an iterative Picard scheme can be expressed as:

$$\phi_k^{(n+1)} = \bar{\phi}_k\,exp\,[\,\mu\sum_i R_{ik}\,(\,Y_i\,/\sum_j R_{i\,j}\,\phi_j^{(n)}\,-1\,)-1\,] \qquad (D.2)$$

For the special case of the uniform a priori probability distribution information (5), Eq.(D.2) reduces to:

$$\phi_k^{(n+1)} = exp\,[\,\mu\sum_i R_{ik}\,(\,Y_i\,/\sum_j R_{i\,j}\,\phi_j^{(n)}\,-1\,)-1\,]\,. \qquad (D.3)$$

REFERENCES

[1] A. Tikhonov and V. Arsenin: *Solution of Ill-posed Problems*. V. H. Winston & Sons, Washington D.C. 1977

[2] R. Kikuchi and B. Soffer: "Maximum Entropy Image Restoration I. The Entropy Expression." J. Opt. Soc. Am., vol.67, no.12, 1656-1665, 1977

[3] J. Skilling and S. Gull: "The Entropy of an Image." SIAM Am. Math. Soc. Proc., vol.14, no.1, 169-189, 1984

[4] L. Shepp and Y. Vardi: "Maximum Likelihood Reconstruction for Emission Tomography." IEEE Trans. Med. Imag., vol.1, no.2, 113-122, 1982

[5] A. Rockmore and A. macovski: "A Maximum Likelihood Approach to Transmission Image Reconstruction from Projections." IEEE Trans. Nucl. Sci., vol.24, no.3, 1929-1935, 1977

[6] B. Frieden: "Restoring with Maximum Likelihood and Maximum Entropy." J. Opt. Soc. Am., vol.62, no.4, 511-518, 1972

[7] B. Frieden: "Statistical Models for the Image Restoration Problem." Comput. Graph. Image Proc., vol.12, no.1, 40-58, 1980

[8] Z. Liang: "Statistical Models of a Priori Information for Image Processing." SPIE Med. Imaging. II, vol.914, no.1, 1988, to appear

[9] C. Shannon: "A Mathematical Theory of Communication." Bell System Tech. J., vol.27, no.3, 379-423, 623-656, 1948; "Communication in the Presence of Noise." Proc. IRE, vol.37, no.1, 10-21, 1949

[10] S. Kullback and R. Leibler: "On Information and Sufficiency." Annals Math. Statistics, vol.22, no.1, 79-86, 1951

[11] E. Jaynes: "Prior Probabilities." IEEE Trans. Sys. Sci. Cyber., vol.4, no.3, 227-241, 1968

[12] D. Luenberger: *Linear and Nonlinear Programming*. Addison-Wesley, Inc, Reading MA, 1984

[13] Z. Liang: "Bayesian Image Processing of Data from Constrained Source Distributions." Ph.D. Dissertation, The City University of New York, 1987

[14] A. Dempster, N. Laird and D. Rubin: "Maximum Likelihood from Incomplete Data via the EM Algorithm." JRSS, vol.39, no.B, 1-38, 1977

[15] E. Isaacson and H. Heller: *Analysis of Numerical Methods*. John Wiley & Sons, Inc., NY, 1966

[16] H. Andrews and B. Hunt: *Digital Image Restoration.* Prentice-Hall, Inc., New York, 1977

[17] C. Rao: *AdvancedStatisticalMethods in Biometric Research.* John Wiley & Sons, Inc., New York, 1952

[18] Z. Liang and H. Hart: "Bayesian Image Processing of Data from Constrained Source Distributions I. non-valued, uncorrelated and correlated constraints." Bull. Math. Biol., vol.49, no.1, 51-74, 1987

[19] K. Lange and R. Carson: "EM Reconstruction Algorithms for Emission and Transmission Tomography." J. Comput. Assist. Tomog., vol.8, no.2, 306-316, 1984

[20] B. Carnahan, H. Luther and J. Wilkes: *Applied Numerical Methods.* John Wiley, New York, 1978

[21] Z. Liang, R. Jaszczak and H. Hart: "Study and Performance Evaluation of Statistical Methods in Image Processing." Submitted for publication

[22] H. Hart and Z. Liang: "Bayesian Image Processing of Data from Constrained Source Distributions II. valued, non-correlated and correlated constraints." Bull. Math. Biol., vol.49, no.1, 75-91, 1987

[23] Z. Liang and H. Hart: "Bayesian Image Processing of Data from Constrained Source Distributions — fuzzy pattern constraints." Phys. Med. Biol., vol.32, no.11, 1481-1494, 1987

[24] Z. Liang and H. Hart: "Source Continuity and Boundary Discontinuity Considerations in Bayesian Image Processing." Med. Physics, 1988, to appear

[25] Z. Liang: "Image Enhancement by Estimated a Priori Information." SPIE Med. Imaging II, vol.914, no.1, 1988, to appear

[26] J. Abbiss, M. Defrise, C. DeMol and H. Dhadwal: "Regularized Iterative and Non-iterative Procedures for Object Restoration in the Presence of Noise: An Error Analysis." J. Opt. Soc. Am., vol.73, no.11, 1983

[27] V. Turchin, V. Koslov and M. Malkevich "The Use of Mathematical-Statistics Methods in the Solution of Incorrectly Posed Problems." Sov. Phys., USPEKHI, vol.13, no.6, 681-840, 1971

[28] G. Herman and A. Lent: "Quadratic Optimization for Image Reconstruction I." Comput. Graph. Image Proc., vol.5, no.2, 319-332, 1976

[29] E. Artzy, T. Elfving and G. Herman: "Quadratic Optimization for Image Reconstruction II." Comput. Graph. Image Proc., vol.11, no.1, 242-261, 1979

[30] G. Minerbo: "MENT: A Maximum Entropy Algorithm for Reconstructing a Source from Projection Data." Comput. Graph. Image Proc., vol.10, no.1, 48-68, 1979

Appendix A: List of Paid Registrants

Ahsanullah M.
Department of D.S. and Computation
Rider College
2083 Lawrenceville Road
Lawrenceville NJ 08648-3099
609-885-5530

Ait-Kheddache Amar
Electrical and Computer Engineering Dept
North Carolina State University
Box 7911
Raleigh NC 27695-7911

Allen Joseph
Rm 4168
USDA, NASS, SRB, R&AD
14th & Independence Ave., SW
Washington DC 20250
(202) 447-3778

Alm Barney
Numerical Algorithms Group Inc.
1101 31st Street, Suite 100
Downers Grove IL 60515
(312) 971-2337

Almond Russell
Department of Statistics
Harvard University
1 Oxford Street, Science Center 609
Cambridge MA 02138
almond@hustat.harvard.edu
(617) 495-4888

Altman Naomi
Cornell University
Ithaca NY

Anandalingam Professor G.
Department of Systems
University of Pennsylvania
Philadelphia PA 19104-6315
anand@eniac.seas
(215) 898-8790

Anneberg L. M.
Wayne State University
3123 Roosevelt
Dearborn MI 48124
(313) 565-2763

Atilzan Taskin
Room 30J-014
AT&T Bell Labs
600 Mountain Avenue
Murray Hill NJ 07974-2070
(201) 582-2824

Atkinson E. Neely
Department of Biomathematics, Box 237
University of Texas System Cancer Center
1515 Holcombe Blvd.
Houston TX 77030
an123651@unhvm1.bitnet
(713) 792-2619

Aubuchon Jay
Minitab, Inc.
3081 Enterprise Drive
State College PA 16801
ATU@PSUVM.BITNET
(814) 238-3280

Ball Celesta G.
5225 Bradfield Drive
Burke VA 22015
(703) 425-5846

Banchoff Thomas
Department of Mathematics
Brown University
Providence RI 02912
(401)863-1129

Banfield Jeff
Department of Mathematical Sciences
Montana State University
Bozeman MT 59717-0001
UMSFJBAN@MTSUNIX1
(406) 994-5367

Baran R.H.
Intellimetrics, Inc.
4508 Cheltenham Drive
Bethesda MD 20814
(301) 394-3264

Barlow Richard
Operations Research Center
University of California
Etcheverry Hall
Berkeley CA 94720
airforce@violet.Berkeley.EDU
(415)642-4993

Barnhill Bruce M.
1014 Farragut Street
Pittsburgh PA 15206

Barr Jewel T.
South Agriculture Building
USDA-NASS
14th & Independence Avenue, SW
Washington DC 20250
(202) 447 7611

Barron Andrew R.
Department of Stat. & Electrical/Com Eng
University of Illinois
725 S. Wright Street
Champaign IL 61820
(217) 333-6216

Barron Austin
Department of Math/Stat
The American University
4400 Massachusetts Avenue
Washington DC 20016
(202) 885-3120

Barron Roger L.
Barron Associates, Inc.
Route 1, Box 159
Stanardsville VA 22973-9511
(804) 985-4400

Barthel Michael
727 Monroe St., #102
Rockville MD 20850
(202) 673-6521

Basinski Antoni
321 Rose Park Drive
Toronto Ontario M4T 1R8
CANADA
(416) 488-2712

Basu Asit
Department of Statistics
University of Missouri-Columbia
328 Math Sciences
Columbia MO 65211
(314) 882-8283

Baxter Ron
SCIRO Division of Math & Stat
PO Box 218
Lindfield NSW 2070
AUSTRALIA
ronb@natmlab.oz.au
61-2-676059

Becker Richard
Statistical & Data Analysis Research
AT&T Bell Labs
600 Mountain Avenue, Rm 2C259
Murray Hill NJ 07974
rab@research.att.com
(201) 582-5512

Bedewi Gabrielle
1023 Brice Road
Rockville Md 20852
(301) 424-1481

Bell James M.
M-233
Milliken & Company
P.O. Box 1926
Spartanburg SC 29304
(803) 573-1627

Benedict Jeffrey P.
8614 Spruce Run Court
Ellicott City MD 21043
(301) 461-7586

Berk Kenneth
Department of Mathematics
Illinois State University
313 Stevenson Hall
Normal IL 61761
(309) 438-8781

Billard Lynne
Department of Statistics
University of Georgia
Athens GA 30602
(404) 542-5232

Birkett Tom
USDA S. Building
Department of Agriculture
14th & Independence, S.W. Rm. 5819
Washington DC 20250
(202) 447-5359

Boggs Paul
Statistical Engineering Division
National Bureau of Standards
Room A37/Admin Building
Gaithersburg MD 20899
Boggs@cam-vax.arpa
(301) 975-3816

Bolorforoush Masood
Center for Computational Statistics
George Mason University
4400 University Drive
Fairfax VA 22030
(703) 764-6181

Bolstein Richard
Center for Computational Statistics
George Mason University
242 Science Technology Building
Fairfax VA 22030
(703) 323-2730

Boudreau Robert M.
1043 Finchley Place
Richmond VA 23225
(804) 257-1301

Box George E. P.
Department of Statistics
University of Wisconsin
1210 West Dayton
Madison WI 53706
(608) 262-2937

Brandenburg Joseph
Intel Scientific Computers
15201 NW Greenbriar Parkway
Beaverton OR 97006
joeb@isc.intel.com
(503) 629-7701

Bravo Maria Soledad
Bureau of the Census
SRD
Washington DC 20233
(301) 763-3950

Brooks Daniel G.
Department of Dec. & Inf Systems
Arizona State University
Tempe AZ 85287-4206
ATDGB@ASUACAD
(602) 965-6350

Bross Neal
Bureau of the Census
Department of Commerce
Rm 3217 FOB #4
Washington DC 20233
(301) 763-4466

Brown Barry W.
Department of Biomathematics, Box 237
University of Texas System Cancer Center
1515 Holcombe Blvd.
Houston TX 77030
an12bwb1@uthvm1.bitnet
(713) 792-2614

Brown C. Hendricks
Department of Biostatistics
Johns Hopkins University
Baltimore MD 21218
(301) 855-2420

Brown Morton B.
Department of Biostatistics
The University of Michigan
109 South Observatory
Ann Arbor MI 48109-2029
(313) 936-0992

Brownstone David
School of Social Sciences
University of California
Irvine CA 92717
(714) 856-6231

Bruce Andrew
395 Cherry Lane, R.R.1
Mendham NJ 07945
deborah@flash.bellcore.com
(201) 879-4319

Bugl Paul
University of Hartford
West Hartford CT 06117
(203) 653-6839

Burg John
Entropic Processing, Inc.
1011 North Foothill Blvd.
Cupertino CA 95014-5649
(408) 973-9800

Burn David A.
BBN Software Products Corporation
10 Fawcett Street
Cambridge MA 02238
(617) 873-4249

Campbell Gregory
LSM, DCRT
National Institute of Health
Bldg. 12A, Rm 3045
Bethesda MD 20892
GGC@NIHCU.BITNET
(301) 496-6037

Carlin Brad
Department of Statistics
University of Connecticut
Box U-120
Storrs CT 06268
(203) 486-4312

Carney John B.
USDA, NASS, RAD, SRB. Rm. 4801
14th & Independence Avenue, SW
Washington DC 20250
(202) 475-3675

Carr Daniel B.
Bettelle Pacific Northwest Laboratories
P.O. Box 999,K6-45
Richland WA 99352
(509) 376-3344

Cassells J. Sally
BBN Software Products
10 Fawcett Street
Cambridge MA 02238
(617) 873-8164

Chaloner Kathryn M.
Department of Applied Statistics
University of Minnesota
352 Classroom-Office Bldg.
St. Paul MN 55108
(612) 625-8722

Chamas N. H.
Wayne State University
7816 Bingham
Dearborn MI 48126
(313) 582-8218

Chambers John M.
AT&T Bell Laboratories
600 Mountain Avenue, Rm. 2C-282
Murray Hill NJ 07974-2070
(201) 582-2681

Chandra Mahesh
Department of Bus. Comp. Inf. Sys.
Hofstra University
201 Computer Center
Hempstead NY 11550
(516) 560-5717

Chandrasekar V.
Department of Electrical Engineering
Colorado State University
Fort Collins CO 80521
ERGU764@CSUGREEN.BITNET
(303) 491-6567

Chari Ravi
Department of Mathematics
Tufts University
Medford MA 02155
rchari@amber.cc.tufts.edu
(617) 628-5000

Chen Zehua
129 Mansfield Street
New Haven CT 06511
(203) 776-0148

Chernoff Herman
Department of Statistics
Harvard University
Cambridge MA 02138
(617) 495-5462

Chin Daniel C.
Applied Physics Laboratory
The Johns Hopkins University
Johns Hopkins Road
Laurel MD 20707-6099
(301) 953-7100

Choo Chanq Y.
Electrical Engineering
Worcester Polytechnic Institute
Worcester MA 01609
cychoo@wpi.bitnet
(617) 793-5000

Chow Mo Suk
Math Department
Northeastern University
Lake Hall
Boston MA 02184
(617) 843-0651

Cibulskis John M.
SPSS, Inc.
444 N. Michigan Avenue
Chicago IL 60611
(312) 329-2400

Ciuffini Mary Ann
Room 4169 So. Bldg
USDA, NASS, SRB RAD
14th and Independence Ave., SE
Washington DC 20250
(202) 447-6751

Clarkson Douglas B.
2500 ParkWest Tower One
IMSL
2500 CityWest Boulevard
Houston TX 77042-3020
(713) 782-6069

Cofer John
200 Stokely Management Center
UTCC
Knoxville TN 37996
Cofer@UTKVXI
(615) 974-6831

Cohen Edgar A.
White Oak
Naval Surface Warfare Center
New Hampshire Avenue
Silver Spring MD 20903-5000
(202)294-2281

Coleman Chuck
2300 Lee Highway, #102
Arlington VA 22201
(703) 527-7699

Conlon Dr. Michael
Department of Statistics
University of Florida
Box J-212, J. Hillis Miller Health Centr
Gainesville FL 32610
(904) 392-2820

Cooper Murray M.
Biomathematics Group 7293-32-2
The Upjohn Company
301 Henrietta Street
Kalamazoo MI 49001-0199

Cox Dennis
Department of Statistics
University of Illinois
725 So. Wright Street
Champaign IL 61820
(217) 333-2972

Crane Giles
Office of Research
NJ State Department of Health
John Fitch Plaza, CN360
Princeton NJ 08625
(609) 292-4315

Creecy Rob
SRD 3215-4
Census Bureau
Washington DC 20233
(301) 763-4466

Crockett Henry D.
Department of Info Syst. and Management
The University of Texas at Arlington
Box 19437
Arlington TX 76019
B718MEE@UPARLVM
(817) 273-3502

Cybenko George
60 Nowell Road
Melrose MA 02176
cybenko@cs.tufts.edu
(617) 381-3214

Dadashzadeh Mohammad
Department of Management Science
University of Detroit
4001 W. McNichols Road
Detroit MI 48221
(313) 927-1237

Davis Linda
11539 Mamie Lane
Fairfax Station VA 22039
(703) 239-0892

de Los Reyes Josephina P.
Department of Mathematical Sciences
University of Akron
Akron OH 44325
(216) 375-7193

Denby Lorraine
AT&T Bell Laboratories
600 Mountain Avenue, Room 2C273
Murray Hill NJ 07922
research!alice!ed
(201) 582-3292

Deng Lih-Yuan
Department of Mathematical Sciences
Memphis State University
Memphis TN 38152
(901) 454-3141

Devlin Thomas F.
Math & Computer Science Department
Montclair State College
Upper Montclair NJ 07043
(201) 893-7244

Dixon Dennis
National Cancer Institute, NIH
Landow Building, Room 4B06
Bethesda MD 20892
(301) 496-4836

Do Kim-Anh
Department of Prevention Research & Bio
Bowman Gray School of Medicine
211-7 Dalewood Drive
Winston-Salem NC 27104
KIM@WFU.EDU
(919) 748-6015

Dombroski Marla
N.I.H.
Federal Bldg, Room 119
Bethesda MD 20892
mpb@nihcu
(301) 496-9725

Don Dr. Allen
77 Mill Spring Road
Manhasset NY 11030
(516) 627-0098

Donnell Deborah
Bell Core Communications Research
435 South Street
Morristown NJ 07960-1961
(201) 829-4319

Dreher Douglas M.
Hughes Aircraft Company
8000 E. Maplewood Avenue, MS CHSB
Englewood CO 80111

Drummey Kevin Ward
National Security Agency
Attn: R51
Fort Meade MD 20755
(301) 859-6461

DuMouchel William
BBN Software Products Corporation
10 Fawcett Street
Cambridge MA 02238
dumouche@SPCA.BBN.COM
(617) 873-8119

DuPont Pierre
BBN Advanced Computers
800 Westpark Drive
McLeane VA 22101
pdupont@bbn.com
(703) 848-4874

Dunn Jeffrey
Code 2842
Naval Research Laboratory
4555 Overlook Avenue, NE
Washington DC 20375-5000
Dunn@NRL.arpa
(202) 767-2884

Duong Dr. Quang Phuc
Bell Canada
1050 Beaver Hall, Room 205
Montreal Quebec H2Z 1S4
CANADA
(514) 870-4923

Durst Mark
Lawrence Livermore National Lab.
L-307, P.O. Box 808
Livermore CA 94550
Durst@clara.llnl.gov
(415) 422-4272

Dykstra Richard L.
Department of Statistics
University of Iowa
Iowa City IA 52242
(319) 335-0823

Eakin Mark
Department of Info Sys & Mgmt Sciences
The University of Texas at Arlington
Box 19437
Arlington TX 76019
B718MEE@UTARLVM1
(817) 273-3502

Eddy William
Department of Statistics
Carnegie-Mellon University
Pittsburgh PA 15213
bill@andrew.cmu.edu
(412) 268-2725

Edwards Don
Department of Statistics
University of South Carolina
Columbia SC 29208
(803) 777-5073

Edwards Lynne K.
323 Burton Hall
University of Minnesota
178 Pillsbury Drive, SE
Minneapolis MN 55455-0211
(612) 624-8381

Efron Bradley
Department of Statistics
Stanford University
Sequoia Hall
Stanford CA 94305
(415) 497-2206

Elder, IV John F.
Barron Associates
Rt. 1 Box 159
Stanardsville NJ 22973-9511
(804) 985-4400

Elfenbein Lowell
TRW
One Federal Systems Park Drive
Fairfax VA 22033
(301) 776-4920

Engelman Laszlo
BMDP Statistical Software, Inc.
1440 Sepulveda Blvd
Los Angeles CA 90025
(213) 479-7799

Ercil Aytul
Math Department
General Motors
General Motors Labs
Warren MI 48098

Ferguson Dania P.
Room 5862-S
USDA-NASS-DMD
1400 Independence Avenue
Washington DC 20250
(202) 447-6690

Feuerverger Andrey
Department of Statistics
University of Toronto
Toronto Ontario M5S 1A1
CANADA
ANDREY@UTSTAT/UUEP
(416) 978-3452

Feygin Leonid
SPSS, Inc.
444 N. Michigan Avenue
Chicago IL 60611
(312) 329-3544

Filliben James
Statistical Engineering Division
National Bureau of Standards
Rm A337/Admin Bldg
Gaithersburg MD 20899
(301) 975-2855

Fischer Martin J.
DCEC, Code R700
1860 Wiehle Avenue
Reston VA 22090-5500
fischer@EDN-VAX.ARPA
(703) 437-2196

Flournoy Nancy
1829 E Capitol St. SE
Washington DC 20003-1711
nflournoy@vote.nsf.gov
(202) 357-3693

Folk Roni
National Cancer Institute
7910 Weedmont Avenue
Bethesda MD 20892
(301) 496-6425

Fong Duncan K. H.
The Pennsylvania State University
340 Beam Business Admin Building
University Park PA 16802
I2V@PSUVM
(814) 863-3541

Franklin Dr. Leroy A.
Department of Systems and Decision Sci.
Indiana State University
Terre Haute IN 47809
(812) 237-2092

Friedman Herman
IBM Technical Education
500 Columbus Avenue
Thornwood NM 10594
(914) 742-5647

Friedman Jerome H.
Stanford Linear Accelerator Center
P.O. Box 4349, Bin 88
Stanford CA 94309
JHF@SLACVM
(415) 926-2256

Fritsvold John D.
Washington Navy Yard
Naval Weapons Engineering Support Act
ESA-31E, Bldg. 220-2
Washington DC 20374-2203
(202) 433-3116

Furnas George
Bell Communications Research
455 South Street
Morristown NJ 07960
gwf@bellcore.com
(201) 829-4289

Gantz Professor Donald T.
Center for Computational Statistics
George Mason University
ST 242
Fairfax VA 22030
(703) 323-2711

Garbo Martin J.
Mail Station CHSB
Hughes Aircraft Company
8000 E. Maplewood Avenue
Englewood CA 80111
(303) 341-3602

Gardenier Dr. T. K.
115 St. Andrews Drive
Vienna VA 22180

Geissler Dr. Paul H.
Patuxent Wildlife Research Center
Route 197
Laurel MD 20708

Gelfand Alan E.
Department of Statistics
The University of Connecticut
196 Auditorium Road, U-120, MSB 428
Storrs CT 06268
GELFAND@UCONN.VM
(203)-486-3416

Gentle James E.
IMSL, Inc.
2500 City West Blvd. Suite 2500
Houston TX 77036
(713) 782-6060

Gerling Thomas
Applied Physics Laboratory
Johns Hopkins Road
Laurel MD 20707

Geweke John
Department of Economics
Duke University
Durham NC 27706
(919) 684-2152

Gilette Robert E.
Department of the Treasury
7217 15th Avenue
Tacoma Park MD 20912
(291) 422-9227

Gilroy Edward J.
U.S. Geological Survey
410 National Center
Reston VA 22029
(703) 648-5714

Gladstein David
119 Gov. Winthrop Road
Somerville MA 02145
(617) 868-2800

Glaz Joseph
Department of Statistics
University of Connecticut
U-120
Storrs CT 06268
(203)486-3413

Goel Prem K.
Department of Statistics
Ohio State University
1958 Neil Avenue
Columbus OH 43210-1247
goel@osupyr.mast.ohiost. edu
(614) 292-8110

Goicoechea Professor Ambrose
Department of Systems Engineering
George Mason University
4400 University Drive
Fairfax VA 22030
(703) 323-3530

Goldman Robert
Simmons College
300 Fenway
Boston MA 02115
(617) 738-2167

Goldstein Larry
Dept of Math, DRB 306 VSC
University of Southern California
University Park - MC 1113
Los Angeles CA 90089-1113
(213) 743-2567

Gray Mary W.
Department of Math Stat
American University
Washington DC 20016
(202) 885-3170

Green Ivan
Statistics Canada

Greene Michael
Math/Stat/Computer Science Department
The American University
4400 Massachusetts Avenue, N. W.
Washington DC 20016
(202) 885-3120

Gu Chong
Department of Statistics
Yale University
Box 2179 Yale Station
New Haven CT 06520
gu-chong@yale.arpa

Gupta Shanti S.
Department of Statistics
Purdue University
W. Lafayette IN 47907
tej@l.cc.purdue.edu
(317) 494-6031

Habib Muhammad
Department of Biostatistics
University of North Carolina
Chapel Hill NC 27514
(919) 966-4887

Hagwood Charles
Statistical Engineering Div
National Bureau of Standards
714-101/A334
Gaithersburg MD 20899
(301) 975-2846

Hahn Dr. Gerald J
GE CRD
P. O. Box 8
Schenectady NY 12301
(518) 387-5558

Hallahan Charles
USDA/ERS
1301 New York Avenue, ND, Rm. 250
Washington DC 20005
(202) 786-1507

Härdle Wolfgang
Department of Economics
University of Bonn
Bonn D-5300
WEST GERMANY
OR624@DBNUOR1.BITNET

Harner James
West Virginia University
215A Morgan Drive
Morgantown WV
(304) 599-9583

Harte James M.
560 N St. SW, N512
Washington DC 20024

Heard Edward
Ciba-Geigy Corporation
556 Morris Avenue
Summit NJ 07901
(201) 277-7085

Hearne Leonard
1829 E. Capital Street, SE
Washington DC 20003
(202) 546-7551

Heckler Charles
Eastman Kodak Company
MQA, B56, Fl 4, KP
Rochester NY 14650
(716) 533-1797

Heiberger Richard M.
Department of Statistics
Temple University
Speakman Hall
Philadelphia PA 19122
(215) 787-6879

Henstridge Dr. John
38 Wellington Street
Oxford OX2 6BB
ENGLAND
0865-510724

Herbert John H.
2929 Rosemary Lane
Falls Church VA 22042
(703) 532-4544

Herzog Thomas N.
U.S.Department of Housing/Urban Dev.
Washington DC 20410-8000

Hietala Paula
Department of Mathematical Sciences
University of Tampere
P.O. Box 607
Tampere SF-33101
FINLAND
358-31-156111

Hill Dr. Mary Ann
BMDP Statistical Software, Inc.
1440 Sepulveda Blvd. Suite 316
Los Angeles CA 90025
(213) 479-7799

Hill Joe R.
EDS Research
2155 Louisiana Blvd, NE Suite 9100
Albuquerque NM 87110
(505) 883-6931

Holland Susan
Suite 500
Mathematica Policy Research Inc.
600 Maryland Avenue, SW
Washington DC 22024
(202) 484-9220

Hormel Phillip
Bldg. D-2078
CIBA/GEIGY Corporation
556 Morris Avenue
Summit NJ 07901
(201) 277-7436

Howe Sally
Scientific Computing Division
National Bureau of Standards
Rm A151 Technology Building
Gaithersburg MD 20899
(301) 975-3807

Hsieh John J.
Department of Preventive Medicine & Bio
University of Toronto
McMurrish Building, 4th floor
Toronto Ontario M5S 1A8
CANADA
(416) 978-5203

Hsu Jason C.
Department of Statistics
The Ohio State University
141 Cockins Hall
Columbus OH 43210
(614) 292-7663

Hudak Gregory B.
Lincoln Center, Suite 106
Scientific Computing Associates
4513 Lincoln Avenue
Leslie IL 60532
(312) 960-1698

Hudson Joseph C.
Department of Science and Mathematics
SMI Engineering and Management Institute
1700 W. Third Avenue
Flint MI 48504

Hurley Catherine
Department of Statistics & Act. Sci.
University of Waterloo
Waterloo Ont N2L 3G1
CANADA
(519) 888-4505

Hwang Jimmie
San Diego State Univ.
Social Science Research Lab
San Diego CA 92042
QALJ002@CALSTATE
(619) 229-2258

Ihaka Ross
Statistics Center
MIT
1 Amherst Street
Cambridge MA 02139
IHAKA@DOLPHIN.MIT.EDU
(617) 253-8404

Inselberg Alfred
IBM Scientific Center
11601 Wilshire Boulevard
Los Angeles CA 90025-1738
(213) 312-5479

Jayachandran Toke
Naval Postgraduate School
Monterey CA 93943
(408) 646-2600

Jernigan Robert W.
Department of Mathematics and Statistics
The American University
4400 Massachusetts Avenue, NW
Washington DC 20016
(202) 885-3120

Johnson Laura
Department of Operations Research
Naval Postgraduate School
Code 55JO
Monterey CA 93943
ljohnson@nps-cs.arpa
(408) 646-2569

Johnson Mark
Statistics Group
Los Alamos National Lab
P.O. Box 1663, MS F600
Los Alamos NM 87545
085290CD90@/SIVAX.LANL.GOV
(505) 667-6334

Johnson Mark A.
MS 7247-267-1
The Upjohn Company
301 Henrietta Street
Kalamazoo MI 49001
(616) 342-9256

Johnstone Iain
Department of Statistics
Stanford University
Sequoia Hall
Stanford CA 94305
iainj@playfair.stanford.edu
(415) 723-9114

Jost Steve D.
Department of Computer Science
DePaul University
243 S. Wabash Avenue
Chicago IL 60604-2302

Kafadar Karen
Hewlett-Packard
1501 Page Mill Road., 4U
Palo Alto CA 94306
kk%hpspd@phlabs.hp.com
(415) 857-7063

Kallianpur Gopinath
Department of Statistics
University of North Carolina
Chapel Hill NC 27514
(919) 962-2187

Kanabar Vijay
Department of Mathematics
University of Winnipeg
515 Portage Avenue
Winnipeg Man. R3B 2E9
CANADA
204 786 9345

Kask Alex W.
28-14 210 Street
Bayside NY 11360
(212) 830-5656

Katzoff Myron
National Center for Health Statistics
3700 East-West Highway
Hyattsville MD 20782
(301) 436-7047

Kauffman Tom
Statistics Department
Rice University
P.O. Box 1892
Houston TX 77251
tom@stat5.rice.edu
(713) 975-1173

Keegel John C.
University of DC
1740 Hobart St. NW
Washington DC 20009
(202) 265-2455

Keller-McNulty Sallie
Department of Statistics
Kansas State University
Dickens Hall
Manhattan KS 66506
(913) 532-6883

Keough Gary
South Building, Rm 4801
USDA/NASS
14th and Independence Avenue, SW
Washington DC 22193
(202) 475-5918

Kettenring Jon R.
BELLCORE
435 South Street, 2Q-326
Morristown NJ 07901
(201) 829-4398

Khan Aqeel A.
Math/Stat Tea, CSCA:RSA
U.S. Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda MD 20814-2797
(301) 295-5566

Khoshgoftaar Taghi M.
Department of Computer Science
Florida Atlantic University
P.O. Box 3091
Boca Raton FL 33431-0991
(305) 393-3855

Kiemele Mark
Air Force Academy
USAFA/DFMS
Colorado Spring CO 80840
(719) 472-4470

Kipnis Victor
Department of Economics
University of Southern California
MC-0035
Los Angeles CA 90089-0035
(213) 743-2487

Kitagawa Genshiro
The Institute of Statistical Mathematics
4-6-7 Minami-Azabu Minato-Ku
Tokyo 106
JAPAN

Knaub, Jr. James
EI-541, U.S. Department of Energy
Energy Information Administration
1000 Independence Avenue
Washington DC 50585
(202) 586-9619

Korin Basil P.
Mathematics Department
American University
Washington DC 20016
(202) 885-3120

LaVarnway Gerard T.
Department of Mathematics
Norwich University
Northfield VT 05663
(802) 485-5011

Lacayo, Jr. Herbert
U.S.E.P.A. (PM-223)
401 M Street
Washington DC 20480
(202) 382-2714

Landauer Christopher
679 Loring Avenue
Los Angeles CA 90024
(213) 336-5635

Launer Robert L.
614 New Kent Place
Cary NC 27511

Le Hung Tri
Center for Computational Statistics
George Mason University
Fairfax VA 22030
(703) 764-6181

LePage Raoul
Department of Statistics & Probability
Michigan State University
East Lansing MI 48824
RDL@LEPAGE-SUN.STT.MSU.EDU
(517) 353-3984

Lebow William
BBN Software Products
10 Fawcett Street
Cambridge MA 02238
lebow@spca@bbn.com
(617) 873-8128

Lenk Peter
New York University
100 Trinity Place
New York NY 10006
(212) 982-2228

Levene Howard
Statistics Department
Columbia University
New York NY 10027
(212) 280-5370

Li Tze Fen
Department of Mathematics
Rutgers University at Camden
Camden NJ 08102
(609) 757-6439

Liggett Walter
National Bureau of Standards
Administration Bldg., A337
Gaithersburg MD 20899
(301) 975-2851

Lii Keh-Shin
Department of Statistics
University of California, Riverside
Riverside CA 92521
(714) 787-3836

Lilliefors Hubert
Statistics Department
George Washington University
Washington DC 20052
HWL1@GWUVM
(202) 994-6664

Lim Dr. Yong Bin
Department of Experimental Statistics
Louisiana State University
Baton Rouge LA 70803
XST720@LSUVM
(504) 388-8383

Lin Charles
2201 Middlefield Court
Raleigh NC 27615
(919) 467-8000

Lin Chih C.
Room 18B-45
FDA, Mail Code HFN-713
5600 Fishers Lane
Rockville MD 20857
(301) 443-4136

Link William
Patuxent Wildlife Research Center
USFWS
Laurel MD 20708

Lippman Alan
Department of Applied Mathematics
Brown University
Box F
Providence RI 02912
(401) 273-4345

Londhe Anil R.
5 Research Parkway
Bristol-Myers Company
P.O. Box 5100
Wallingford CT 06492
(203) 284-6241

Low Leone
Math and Statistics Department
Wright State Unviversity
Dayton OH 45435
(513) 873-2256

Lucke Joseph
Lehigh Valley Hospital Center
1200 S. Cedar Crest Blvd.
Allentown PA 18103
(215) 776-8889

Maar Dr. James R.
National Security Agency
Attn: R51
Fort Meade MD 20755-6000
(301) 859-6341

MacGibbon-Taylor Brenda
Department of Decision Science
Concordia University
1455 de Maisonneuve Blvd. W
Montreal QUE H3G 1M8
Canada
(514) 848-2982

Makris Nick
Gillette Research Institute
1413 Research Boulevard
Rockville MD 20850
(301) 738-0272

Makuch William M.
Research and Development Center
General Electric-CRD
P.O. Box 8 Kl-4C41A
Schenectady NY 12301
ARPANET: MAKUCH@GE-CRD
(518) 387-5918

Manchester Lise
Department of Math, Statistics & CS
Dalhousie University
Halifax Nova Scotia B3H 3J5
CANADA
lise@dalcs.uucp
(902) 424-3624

Mann Nancy R.
Department of Biomathematics
University of California
Los Angeles CA 90024

Mathieu Claire
Computer Science Department
Princeton University
Princeton NJ 08544
(609) 452-5496

Matthews Peter
Department of Mathematics and Statistics
University of Maryland, Baltimore County
Baltimore MD 21228
(301) 455-2423

Mazur Catherine
USDA-NASS
Room 4801 S. Bldg.
Washington DC 20250
(202) 475-3483

Mazzuchi Thomas A.
Department of Operations Research
George Washington University
Washington DC 20052
(703) 994-7514

McClave Jim
P.O. Box 14545
Gainesville FL 32605
(904) 375-7624

McClure Don E.
Department of Applied Mathematics
Brown University
Box F
Providence RI 02912
dem@brownvm.bitnet
(401) 863-1496

McDonald Bill
Naval Surface Warfare Center
White Oak Lab
Silver Spring MD 20903-5000
(202) 394-2585

McDonald John Alan
Statistics, GN-22
University of Washington
Seattle WA 98195
JAM
@ENTROPY.MS. WASHINGTON.EDU
(206) 545-7438

McIntosh Allen
Room 2C334
Bell Communications Research
435 South Street,
Morristown NJ 07960-1961

McKenzie, Jr. John D.
Babson College
Babson Park
Wellesley MA 02157-0901
(617) 239-4479

McKinney Steve
Department of Biostatistics SC-32
University of Washington
Seattle WA 98195

McLeish D. L.
University of Waterloo
Waterloo Ontario N2L 2V5
CANADA

McLeish Mary
Department of Computing and Information
University of Guelph
Geulph Ontario N1G 2W1
CANADA
(519) 824-4120

Mehra Munish
Department of Statistics
University of Kentucky
Lexington KY 40506
munish@ukma
(606) 257-4423

Mellor-Crummy John
Department of Computer Science
University of Rochester
Rochester NY 14627
crummey@rochester.edu
(716) 275-0922

Mergerson James W.
USDA, NASS, RAD, SSB
3251 Old Lee Highway, Rm. 506
Fairfax VA 22030

Messer Karen
Department of Mathematics
California State University
Fullerton CA
(714) 774-3631

Meyer Michael
Statistics Department
Carnegie-Mellon University
Pittsburgh PA 15213
MM8S@andres.cmv.edu
(412) 268-3108

Meyer Ruth K.
Department of Business Computer Inf. Sys
St. Cloud State University
St. Cloud MN 56301
(612) 255-2241

Mikhail Nabih
Department of Mathematics
Liberty University
Box 20000
Lynchburg VA 24506-8001
(804) 237-5961

Millar P.W.
Department of Statistics
University of California, Berkeley
Berkeley CA 94720
(415) 642-2781

Miller John
Center for Computational Statistics
George Mason University
4400 University Drive
Fairfax VA 22030
(703) 323-2733

Miller Michael F.
Hoechst-Roussel Pharmaceuticals Inc.
Route 202-206N, Bldg M
Somerville NJ 08876
(201) 231-3486

Minor James M.
Du Pont Company
P.O. Box 6090
Newark DE 19714-6090
(302) 366-2432

Mitchell Toby
Math Stat Research, Eng Physics Division
Oak Ridge National Laboratory
P.O. Box Y, Bldg. 9207A, MS3
Oak Ridge TN 37831
mitchell@msr.epm.ornl.gov
(615) 574-3143

Mittnik Stefan
Department of Economics
State University of NY at Stony Brook
Stony Brook NY 11794-4384
SMITTNIK@SBCCVM (Bitnet)
(516) 632-7532

Miyashita Dr. Junryo
Department of Computer Science
California State University
5500 University Parkway
San Bernardino CA 92373
(714) 887-7647

Modarres Reza
Mathematics, Statistics and CS Dept.
The American University
Massachusetts Avenue
Washington DC 20016
(202) 885-3149

Mode Dr. Charles J.
Department of Math. and Computer Sci.
Drexel University
Philadelphia PA 19104
215-895-2668

Mohns Cynthia
Numerical Algorithms Group Inc.
1101 31st Street, Suite 100
Downers Grove IL 60515
(312) 971-2337

Mokatrin Ahmad
Math/Stat Department, Clark Hall
American University
4400 Massachusetts Avenue
Washington DC 20010

Moore Marc
Ecole Polytechnique Montreal
C.P. 6079 Succursale "A"
Montreal Quebec H3C 3A7
CANADA
(514) 340-4513

Moser Barry
Department of Exper. Statistics
Louisiana State University
Ag. Administration Building
Baton Rouge LA 70803-5606
XST769@LSUVM.bitnet
(504) 388-8303

Moussa-Hamouda Effat
DePaul University
2323 N. Seminary, #572-C
Chicago IL 60614
(312) 341-8250

Muller Mervin E.
Department of Computer & Inf. Science
The Ohio State University
2036 Neil Avenue Mall
Columbus OH 43210-1277
(614) 292-5973

Munson Peter J.
National Institutes of Health
Building 10, Room 6C101
Bethesda MD 20892
PTM@NIHCUDEC
(301) 496-2972

Nadas Arthur
IBM Research
P.O. Box 218
Yorktown Height NY 10598
NADAS@YKTVMX
(914) 945-2163

Nelder J. A.
Imperial College
Huxley Bldg., 180 Queens Gate
London SW7 2BZ
ENGLAND
UK+1-589-5111

Newton H. Joseph
Department of Statistics
Texas A&M University
College Station TX 77843-3143
(409) 845-3141

Nguyen Trung H.
EIC Labs
111 Downey Street
Norwood MA 02062
(617) 769-9450

Nicoll Jeff
IDA
1801 N. Beauregarde Street
Alexandria VA 22311
(703) 478-2987

Nolan James R.
Quantitative Business Analysis
Siena College
Loudonville NY 12211
(518) 783-2503

Normolle Daniel
Department of Biostatistics
University of Michigan
School of Public Health
Ann Arbor MI 48109-2029
313-936-1004

Nychka Douglas
Department of Statistics
North Carolina State University
Box 8203
Raleigh NC 27695-8203
NYCHKA@NCSUSTAT.BITNET
(919) 737-2534

O'Brien Fanny
BBN Software Products
10 Fawcett Street
Cambridge MA 02238
(617) 497-3778

O'Connor Carol
EMACS Department
University of Louisville
Speed School
Louisville KY 40292
(502) 588-6304

O'Connor Thomas A.
O'Connor and Associates, Inc.
3017 Juniper Hill Road
Louisville KY 40206

Oldford R. Wayne
Department of Statistics & Act. Sci
University of Waterloo
Waterloo Ontario N2L 3G1
CANADA
RWOLDFORD@
WATER.WATERLOO.EDU
(519) 888-4609

Ondrasik John A.
P.O. Box 368
Boehringer Ingelheim Pharmaceuticals
90 East Ridge
Ridgefield CT 06877
(203) 798-4243

Ostrouchov George
Mathematical Sciences Section
Oak Ridge National Laboratory
Building 9207A MS-3
Oak Ridge TN 37831
OST@MSR.EPM.ORNL.GOV
(615) 574-3137

Owen Art
Department of Statistics
Stanford University
Sequoia Hall
Stanford CA 94305
art@playfair.stanford.edu
(415) 725-2232

Ozga Martin
Room 4168
USDA, NASS, RAD, SRB
14th & Independence Ave., SW
Washington DC 20250
(202) 447-5483

Pacheco Nelson S.
1259 Lake Plaza Drive
Colorado Spring CO 80906
(719) 550-6376

Padgett William J.
Department of Statistics
University of South Carolina
Columbia SC 29208
(803) 777-5070

Park Tae-woo
AFAL/SCS
Edwards AFB CA 93523-5000
(805) 275-5196

Parzen Emanuel
Department of Statistics
Texas A&M University
College Station TX 77843-3143
(409) 845-3188

Patterson David
Department of Math Sciences
University of Montana
Missoula MT 59812
(406) 243-6748

Peck Roger W.
Department of Computer Science and Stat.
The University of Rhode Island
Tyler Hall
Kingston RI 02881-0816
(401) 792-4497

Pederson Shane
P.O. Box 1663
Los Alamos National Laboratory
Mail Stop F-600
Los Alamos NM 87544
100176A26@S1VAX.LANL.GOV
(505) 667-7303

Pei Gabriel P.
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria VA 22311
(703) 578-2882

Peierls Ronald F.
Applied Mathematics Department - 515
Brookhaven National Laboratory
Upton NY 11973
peierls@bnl.bitnet
(516) 282-4104

Percival Don
Applied Physics Lab
University of Washington
Seattle WA 98105
(206) 543-1300

Perry Charles R.
USDA/NASS
14th & Independence Avenue S.W.
Washington DC 20250
(202) 475-3075

Peruggia Mario
Department of Statistics
Carnegie-Mellon University
232 BH
Pittsburgh PA 15213
(412) 268-8590

Pesek John
College of Agricultural Sciences
University of Deleware
Department of Food & Resource Economics
Newark DE 19717-1303
FOD31626@UDACSVM bitnet
(302) 451-1319

Phillips Abraham
Research Dept
Prudential Property & Cas Insurance Co.
23 Main Street
Holmdel NJ 07733
(201) 946-5109

Phinney Stephen E.
250/060
IBM Manassas
9500 Godwin Drive
Manassas VA 22110
(703) 367-2403

Pickle Dr. Linda W.
NCI
Landon Building, Room 3A06
Bethesda MD 20892
(301) 496-6425

Pieper Carl
295 Central Park West, 15D
New York NY 10024
(212) 799-8212

Pierce Alan
Amoco Production Company
4502 East 41st Street
Tulsa OK 74136
(918) 660-3830

Pierce Margaret Anne
Georgia Southern College
LB 8093
Statesboro GA 30460
(912) 681-5427

Pierchala Carl E.
Food & Drug Administration
P.O. Box 1554
West Bethesda MD 20817
(301) 443-4594

Pitcher Hugh M.
US - EPA
3031 Beech Street, N. W.
Washington DC 20015
(202) 382-2788

Pregibon Daryl
AT&T Bell Labs
600 Mountain Avenue, Room 2C-264
Murray Hill NJ 07974
(201) 582-3193

Puri Madan
Department of Mathematics
Indiana University
Bloomington *IN 47405*

Raj Baldev
School of Business and Economics
Wilfrid Laurier University
Waterloo Ont N2L 3C5
CANADA
(519) 884-1970

Ramirez Donald E.
Department of Mathematics
University of Virginia
Math-Astro Building
Charlottesville VA 22903
DER@VIRGINIA
(804) 924-4934

Ratnaparkhi M.V.
Department of Mathematics and Statisites
Wright State University
Dayton OH 45435
(513) 873-2193

Raubertas Richard F.
Department of Statistics
University of Rochester
Rochester NY 14627
(716) 275-2406

Rayens William S.
Department of Statistics
University of Kentucky
859 Patterson Office Tower
Lexington KY 40506-0027
(606) 257-7061

Richardson David
10211 Brunswick Ave.
Silver Spring MD 20902
(301) 649-2650

Richter Don
Graduate School of Business Admin.
New York University
100 Trinity Place
New York NY 10006
(212) 285-6130

Ringeisen Richard D
Department of Mathematical Sciences
Clemson University
Clemson SC 29634-1907
RDRNG@Clemson
(803) 656-5245

Rom Dror
441 Tomlinson Road #F-13
Philadelphia PA 19116
(215) 661-6336

Rosenblatt Joan R.
National Bureau of Standards
Administration bldg, Room A438
Gaithersburg MD 20899
975-2733

Rovine Michael J.
Department of Individual and Family Stud
Penn State
S-110 Henderson Building
University Park PA 16802
(814) 863-0267

Rumpf David L.
Department of Ind. Eng. and Inf. Systems
Northeastern University
360 Huntington Avenue, Rm 330 Snell
Boston MA 02115
(617) 437-3632

Ruskin David
Center for Naval Analyses
P.O. Box 16268
Alexandria VA 22302
(703) 824-2284

Russell Carl T.
U.S. Army OTEA, CSTE-TS-R
5600 Columbia Pike
Falls Church VA 22041
(703) 756-1818

Rust John
University of Wisconsin
1180 University Drive
Madison WI 53706
(608) 263-3871

Ryan Barbara
Statistics Department
Stanford University
Sequoia Hall
Stanford CA 94305-4065
BFR@PLAYFAIR.STANFORD.EDU
(415) 723-2787

Ryan Thomas A.
Minitab, Inc.
3081 Enterprise Drive
State College PA 16801
(814) 238-3280

Salomon Matthew A.
Fiscal Analysis Division
Congressional Budget Office
2nd & D Streets, SW
Washington DC 20515
(202) 226-2765

Samuels Stephen M.
Department of Statistics
Purdue University
W. Lafayette IN 47907
(317) 494-6042

Samuelson Douglas A.
International Telesystems Corporation
600 Herndon Parkway
Herndon VA 22070

Sawyer John W.
Texas Tech University
1A104 Health Sciences Center
Lubbock TX 79430
(806) 743-2146

Scharff H. Felix
Dept 49WA, Stu 260
IBM
Neighborhood Road
Kingston NY 12401
(914) 385-4013

Schiller Susannah
5 Scharlet Sage Court
Burtonsville MD 20866
Schill@NBS
(301) 776-1187

Schmee Josef
Union College
Bailey Hall 311
Schenectady NY 12308
schmeej@union
(518) 370-6248

Schmeiser Bruce W.
School of Industrial Engineering
Purdue University
Grissom Hall
West Lafayette IN 47907
schmeise@pink.ecn.purdue.edu
(317) 494-5422

Schwemberger John
8154 Larkin Lane
Vienna VA 22180
(202) 382-7195

Scott David
Department of Statistics
La Trobe University
Bundoora Victoria 3083
AUSTRALIA
STADTS@latvaa8.lat.amurnari.oz
(61)3 4792091

Scott David W.
Department of Statistics
Rice University
P.O. Box 1892
Houston TX 77251-1892
(713) 527-8101

Scott Dr. David
Dept of Decision Sciences and Management
Concordia University
1455 de Maisonneuve Blvd. West
Montreal Quebec H3G 1M8
CANADA
(514) 848-2969

Seaman John W.
Department of Statistics
University of Southwestern Louisiana
P.O. Box 41006
Lafayette LA 70506-1006
(318) 231-5294

Segall Dr. Richard
Department of Mathematics
University of Lowell
Olsen Hall
Lowell MA 01854
(617) 452-5000

Sessions Dr. David N.
9 Purdue Road
Glen Cove NY 11542
(516) 676-2123

Shanmugam Ron
University of Colorado
Denver CO 80204
(303) 556-8463

Shiau Jyh-Jen Horng
Department of Statistics
University of Missouri-Columbia
222 Mathematical Science Bldg
Columbia MO 65211
(314) 882-7467

Shiau Tzong-Huei
Department of Computer Science
University of Missouri-Columbia
Columbia MO 65211
csshiau@umcvmb.bitnet
(314) 882-4540

Shier Douglas R.
Department of Mathematics
College of William and Mary
Williamsburg VA 23185
$DRSHIE@WMMVS
(804) 253-4481

Shing Chen-Chi
Department of Computer Science
Radford University
Box 5752
Radford VA 24142
(703) 831-5733

Shrager Richard I.
National Institutes of Health
Bldg 12A, Room 2041
Bethesda MD 20892
(301) 496-1122

Silverberg Arthur
Food and Drug Administration
4600 Fishers Lane, HFV-124
Rockville MD 20857
(301) 443-1580

Simon Dr. Richard
Room 4B06
National Cancer Institute
7910 Woodmont Avenue
Bethesda MD 20892
(301) 496-4836

Singpurwalla N.D.
Department of Operations Research
George Washington University
Washington DC 20052
(202) 994-7515

Siu Cynthia O.
Osler 622
The Johns Hopkins University
600 N. Wolfe Street
Baltimore MD 21205

Sleeper Lynn A.
Department of Biostatistics
Harvard School of Public Health
677 Huntington Avenue
Boston MA 02115
(617) 732-3626

Slinkman Craig W.
Department of Inf Sys & Mgmt Sciences
The University of Texas at Arlington
Arlington TX 76019
B718MEE@UPARLVM1
(817) 273-3502

Slowinski Samuel M.
Federal Reserve Board
20th & Constitution Avenue, NW
Washington DC 20551
(202) 452 2622

Smith Adrian
Department of Mathematics
University of Nottingham
Nottingham NG7 2RD
UNITED KINGDOM
602 484848

Smith Eric P.
Department of Statistics
Virginia Polytechnic Institute & State U
Blacksburg VA 24061
(703) 961-7932

Smith Laurie Melany
9521 Baltimore Avenue
Laurel MD 20707
(301) 490-9665

Snell Robert
Eastman Kodak Company
MQA, Bldg 56, Fl 4, KP
Rochester NY 14650

Sofer Ariela
Operations Research and Applied Statistics
George Mason University
4400 University Drive
Fairfax VA 22030
(703) 323-2728

Somerville Paul N.
University of Central Florida
P.O. Box 25000
Orlando FL 32816
(305) 275-2695

Song Wheyming Tina
Purdue University
West Lafayette IN 47906
(317) 698-7961

Sood Arun K.
Department of Computer Science
George Mason University
4400 University Drive
Fairfax VA 22030
(703) 323-3395

Soyer Refik
Department of OR
George Washington University
Washington DC 20052
(202) 994-6794

Speckman Paul
Department of Statistics
University of Missouri-Columbia
Columbia MO 65211
(314) 882-7783

Stephenson Elizabeth
P.O. Box 7375
2375 Garcia Avenue
Moutain View CA 94039
(415) 960-7784

Stern Hal
Department of Statistics
Harvard University
One Oxford Street
Cambridge MA 02138

Stewart G. W.
University of Maryland
College Park MD 20742
stewart@thales.cs.umd.edu
(301) 454-6120

Stewart Leland
Department 92-20, Bldg. 254E
Lockheed Research Laboratory
3251 Hanover Street
Palo Alto CA 94304
(415) 424-2710

Studdiford Walter B.
Registrar's Office
Princeton University
B10 A West College
Princeton NJ 08544
(609) 452-6195

Stuetzle Werner
Department of Statistics
University of Washington
GN-22
Seattle WA 98195
(206) 543-4386

Sutton Cliff
Center for Computational Statistics
George Mason University
242 Science Technology Building
Fairfax VA 22030
csutton@gmuvax
(703) 323-3863

Szewczyk William F.
2905 Shamrock Terrace
Olney MD 20832
(301) 774-1158

Takane Yoshio
Department of Psychology
McGill University
1205 Dr. Penfield Avenue
Montreal Quebec H3A 1B1
CANADA
PS81@MCGILLA
(514) 398-6125

Tarter Michael E.
Department of Statistics
University of California
32 Earl Warren Hall
Berkeley CA 94720
(415) 642-4601

Tasker Gary D.
U.S. Geological Survey
430 National Center
Reston VA 22092
(703) 648-5892

Tawfik Lorraine
40 Amityville Street
Islip Terrace NY 11752
(516) 277-2875

Taylor D. Wayne
Department of Clinical Epidemiology&Bio.
McMaster University
Health Sciences Center
Hamilton Ontario L8N 3Z5
CANADA
525-9140 X4102

Teitel Robert F.
Teitel Data Systems
7200 Wisconsin Avenue, Suite 410
Bethesda MD 20814
(301) 656-0401

Terpenning Irma
Rd2 Box 109
Frenchtown NJ 08825
(201) 582-2268

Therneau Terry M.
May Clinic
200 First Street SW
Rochester NM 55905
(507) 284-8803

Thisted Ronald A.
Department of Statistics
University of Chicago
5734 University Avenue
Chicago IL 60637
thisted@galton.uchicago.edu
(312) 702-8333

Thompson James R.
Department of Statistics
Rice University
P.O. Box 1892
Houston TX 77251-1892
(713) 527-4828

Thornton Ding H.
Naval Air Test Center
Computer Sciences Directorate
Patuxent River MD 20670-5304
(301) 863-3396

Tierney Luke
School of Statistics
University of Minnesota
Minneapolis MN 55455
luke%umnstat@umn-cs.arpa
(612) 625-7843

Tretter Dr. Marietta
Business Analysis & Research
Texas A.& M. University
College Station TX 77843
(409) 845-1383

Tsong Yi
FDA HFN-715
6500 Fisher Lane
Rockville MD
(301) 443-4710

Tsou Tai-Houn
3637 Canyon Crest Drive, A307
Riverside CA 92507
(714) 788-4656

Tubb Gary
Instructional Computing
University of South Florida
USF 3185
Tampa FL 33620
DNPABAA@CFRVM

Tukey Paul A.
Bell Communications Research
435 South Street
Morristown NJ 07960
(201) 829-4285

Turner David L.
Department of Mathematics & Statistics
Utah State University
Logan Utah 84322-3900
DTURNER@USU
(801) 750-2814

Unger Elizabeth
Computing and Information Sciences
Kansas State University
243 Nichols Hall
Manhattan KS 66506

Utts Jessica
SRI
333 Ravenswood Avenue
Menlo Park CA 94025
utts@unix.sri.com
(415) 859-4445

Varner Ruth
369 Holmes Drive
Vienna VA 22180
(703) 938-9209

Varty John Franklin
6602 Boulevard View Place
Alexandria VA 22307
(703) 765-0540

Venetoulias Achilles
E40-133
MIT, Sloan School
1 Amherst Street
Cambridge MA 02139
axilleas@dolphin.mit.edu
(617) 253-8416

Vernhes Frederique L.
Department of Statistics
Yale University
Box 2179, Yale Station
New Haven CT 06511
(203) 782-0430

Vetter John E.
Washington Navy Yard
Naval Weapons Engineering Support Act
ESA-31, Bldg. 220-2
Washington DC 20374-2203
(202)433-3621

Vitter Jeffrey S.
Department of Computer Science
Brown University
Box 1910
Providence RI 02904
jsv@cs.brown.edu
(401) 863-3300

Von Eye Alexander
Department for Individual Family Study
The Pennsylvania State University
University Park PA 16802
(814) 863-0267

Waclawiw Myron
5364 Hesperus Drive
Columbia MD 20144
(301) 730-0294

Wahba Grace
Department of Statistics
Yale University
Box E2179 Yale Station
New Haven CT 06520-2179
wahba@celray.cs.yale.edu
(203) 432-0666

Walker Homer
Department of Mathematics
Utah State University
Logan UT 84322-3900
uf7099@usu.bitnet
(801) 750-2026

Wang Chaiho
1232 Meyer Court
McLean VA 22101
(202) 724-6368

Wang R. H.
P.O. Box 586
OLIN
350 Knotter Drive
Cheshire CT 06410
(203) 271-4196

Weber James S.
Department of Management
Roosevelt University
P.O. Box 603
Gurnee IL 60031-0603

Wegman Edward J.
Center for Computational Statistics
George Mason University
242 Science Technology Building
Fairfax VA 22030
ewegman@gmuvax.gmu.edu
703 323 2723

Weidman Scott
MRJ, Inc.
10455 White Granite Drive
Oakton VA 22124
(703) 385-0879

Weiss Guenter
University of Winnipeg
515 Portage Avenue
Winnipeg Man R3B 2E9
CANADA
(204) 786-9399

Weiss Robert E.
Department of Applied Statistics
University of Minnesota
Classroom Office Bldg. 352
St. Paul MN 55108
weiss@umnstat.stat.umn.edu
(612) 625-2756

Welsch Roy E.
M.I.T.
50 Memorial Drive, E53-383
Cambridge MA 02139
(617) 253-6601

Wesley Robert
Department of Health and Human Services
9807 Owen Brown Road
Columbia MD 21045
(301) 496-7946

Whitney David A.
TASC
55 Walkers Brook Drive
Reading MA 01867
(617) 942-2000

Whitridge Patricia
Business Survey Methods Division
Statistics Canada
RH Coats Bldg 11-C, Tunney's Pasture
Ottawa Ontario K1A 0T6
CANADA
(613) 951-8614

Wilburn Arthur J.
4600 Jasmine Drive
Rockville MD 20853-1737
(301) 929-1040

Wilson P. David
504 Shadow Grove Court
Lutz FL 33549
(813) 974-4860

Winkler Gernot
Time Service Department
U.S. Naval Observatory
34th & Massachusetts Avenue, NW
Washington DC 20392-5100
(202) 653-1520

Winkler William E.
Census Bureau
Washington DC 20233
(301) 763-3905

Wochnik Michael
1212 Gibbon #6
Laramie WY 82070
(307) 745-9393

Wollan Peter
Department of Mathematics
Michigan Technological University
Houghton MI 49931
USA
(906) 487-2694

Wolting Duane
Aerojet TechSystems Company
P.O. Box 13222, Bldg. 2002, Dept. 9470
Sacramento CA 95813
(916) 355-2692

Woodburn Rose Louise
8426 Ravenswood Road
New Carrollton MD 20784
(301) 459-5138

Woodfield Terry J.
SAS Institute Inc.
SAS Circle, Box 8000
Cary NC 27512-8000
(919) 467-8000

Woodruff Brian
Bolling AFB
AFOSR
Washington DC 20332
(202) 767-5027

Wyscarver Roy A.
Economic Modeling &
Computer Application
U.S. Treasury Department
15th & Pennsylvania Ave., NW
Washington DC 20220
(202) 566-5085

Yang C. C.
NRL
Code 5380
Washington DC 20375

Yang Ting
University of Cincinnati
ML 025
Cincinnati OH 45221
(513) 475-5619

Young Dean M.
Department of Information Systems
Baylor University
Waco TX 76798
(817) 755-2258

Youngren Mark A.
3809 Terrace Drive
Annandale VA 22003
(202) 295-1625

Yu Chen Cheng W.
Dept. 23W, Bldg. 630, E60
IBM Corporation - East Fishkill
Route 52
Hopewell Junc. NY 12533-0999
(914) 892-2200

Appendix B: Author Index